

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
Кафедра автоматизованих систем обробки інформації і управління

До захисту допущено:

В.о. завідувача кафедри

(підпис) Олександр ПАВЛОВ
(вл.ім'я, прізвище)

“ ____ ” _____ 2020 р.

Дипломний проєкт
на здобуття ступеня бакалавра

**за освітньо-професійною програмою «Інформаційні управляючі
системи та технології»
спеціальності 122 «Комп'ютерні науки та інформаційні технології»**

на тему: «Підсистема продажу та обліку фармацевтичних
препаратів корпоративної ERP системи»

Виконала:

студентка IV курсу, групи ІС-361

Гаєвська Аліна Олегівна

(прізвище, ім'я, по батькові)

(підпис)

Керівник

ст.викл. Халус Олена Андріївна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

**Консультант з
графічної
документації**

ст.. викл. Проскура Світлана Леонідівна

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, прізвище, ім'я, по батькові)

(підпис)

Засвідчую, що у цьому дипломному проєкті
немає запозичень з праць інших авторів без
відповідних посилань.

Студент (-ка) _____
(підпис)

Київ – 2020 року

Національний технічний університет України
“Київський політехнічний інститут імені Ігоря Сікорського”

Факультет (інститут) інформатики та обчислювальної техніки
(повна назва)

Кафедра автоматизованих систем обробки інформації і управління
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 122 «Комп'ютерні науки та інформаційні технології»

Освітньо-професійна програма «Інформаційні управляючі системи та технології»

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

Олександр ПАВЛОВ
(підпис) (вл.ім'я, прізвище)

“ ” 2020 р.

ЗАВДАННЯ
на дипломний проєкт студенту

Гаєвській Аліні Олегівні

(прізвище, ім'я, по батькові)

1. Тема проєкту «*Підсистема продажу та обліку фармацевтичних препаратів корпоративної ERP системи*»

керівник проєкту Халус Олена Андріївна, ст.викл.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від “7”травня 2020 р. №1081-с

2. Термін подання студентом проєкту “01”червня 2020 року

3. Вихідні дані до проєкту

Технічне завдання

4. Зміст пояснювальної записки

1. Загальні положення: основні визначення та терміни, опис предметного середовища, огляд ринку програмних продуктів, постановка задачі

2. Інформаційне забезпечення: вхідні дані, вихідні дані, опис структури бази даних

3. Математичне забезпечення: змістовна та математична постановки задачі, обґрунтування та опис методу розв'язання

4. Програмне та технічне забезпечення: засоби розробки, вимоги до технічного забезпечення, архітектура програмного забезпечення, побудова звітів

5. Технологічний розділ: керівництво користувача, методика випробувань програмного продукту

5. Перелік графічного матеріалу

1. Схема структурна варіантів використань
2. Схема бази даних
3. Схема структурна послідовності
4. Схема структурна класів програмного забезпечення
5. Схема структурна компонентів
6. Креслення вигляду екранних форм

6. Консультанти розділів проєкту

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання «13» квітня 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1.	<i>Вивчення рекомендованої літератури</i>	<i>15.04.2019</i>	
2.	<i>Аналіз існуючих методів розв'язання задачі</i>	<i>20.04.2019</i>	
3.	<i>Постановка та формалізація задачі</i>	<i>27.04.2019</i>	
4.	<i>Розробка інформаційного забезпечення</i>	<i>05.04.2019</i>	
5.	<i>Алгоритмізація задачі</i>	<i>05.04.2019</i>	
6.	<i>Обґрунтування використовуваних технічних засобів</i>	<i>12.04.2019</i>	
7.	<i>Розробка програмного забезпечення</i>	<i>06.05.2019</i>	
8.	<i>Налагодження програми</i>	<i>07.05.2019</i>	
9.	<i>Виконання графічних документів</i>	<i>08.05.2019</i>	
10.	<i>Оформлення пояснювальної записки</i>	<i>11.05.2019</i>	
11.	<i>Подання ДП на попередній захист</i>	<i>18.05.2019</i>	
12.	<i>Подання ДП на основний захист</i>	<i>01.06.2020</i>	
13.	<i>Подання ДП рецензенту</i>	<i>02.06.2020</i>	

Студент

Аліна ГАЄВСЬКА

Керівник

Олена ХАЛУС

[illegible]

Пояснювальна записка

до дипломного проєкту

на тему: Підсистема продажу та обліку фармацевтичних препаратів

корпоративної ERP системи

Київ – 2020 року

АНОТАЦІЯ

Структура та обсяг роботи. Пояснювальна записка дипломного проекту складається з шести розділів, містить 10 рисунків, 8 таблиць, 1 додаток, 6 джерел.

Дипломний проект присвячений розробці веб-застосування для автоматизації процесів аптечного закладу. У дипломному проекті розроблено веб-сайт для підтримки діяльності аптечного закладу.

Розділ загальних положень включає в себе опис предметного середовища, огляд наявних аналогів та постановку задачі. Також цей розділ включає опис діяльності та функціональних складових системи.

У розділі інформаційного забезпечення були визначені вхідні та вихідні дані, описана структура бази даних.

Розділ математичного забезпечення присвячений обрахунку статистичної інформації на основі виданих лікарських засобів у аптечному закладі.

У розділі програмного забезпечення вказано діаграми послідовності, класів, компонентів, специфікацію функцій. Також представлено опис архітектури системи, програмного та апаратного забезпечення.

У технологічному розділі наведено керівництво користувача, опис випробувань веб-застосування.

ПІДСИСТЕМА, АПТЕЧНИЙ ЗАКЛАД, РЕЦЕПТ, ФАРМАЦЕВТ,
ВЕБ-ЗАСТОСУВАННЯ, ПОГАШЕННЯ, ІНТЕРНЕТ

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		2

ABSTRACT

Structure and scope of work. Guidance note of degree project consists of six chapters, contains 10 pictures, 8 tables, 1 application, 6 sources.

Diploma project is dedicated to the development of the web-application for pharmacy. In this project was developed web-site for supporting pharmacy activity and dispensing recipes.

General terms section includes description of the objective environment, review of the existing analogues and statement of the problem. In addition, this section includes description of activity and functional components of the system.

In the information support section input and output data was identified and database structure is shown.

Mathematical section is dedicated to the development of statistics based on the dispensed recipes.

The software section includes sequence, class, component diagrams and function specification. There is a description of development tools and overview of architecture, software and hardware tools.

Technological section provides the user manual and description of web-application.

PHARMACY, RECIPE, PHARMACIST. WEB-APPLICATION,
DISPENSING, INTERNET.

					ДП ІС-6223.00.000.ПЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

ЗМІСТ

ВСТУП	6
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ	7
1.1 Опис предметного середовища	7
1.1.1 Опис процесу діяльності	8
1.2 Огляд наявних аналогів	13
1.3 Постановка задачі	15
1.3.1 Призначення розробки	15
Висновок до розділу	16
2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ	17
2.1 Вхідні дані	17
2.2 Вихідні дані	17
2.2.1 Опис структури бази даних	17
Висновок до розділу	22
3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ	23
3.1 Змістовна постановка задачі	23
3.2 Математична постановка задачі	23
3.3 Обґрунтування методу розв'язання	23
3.4 Опис методів розв'язання	24
Висновок до розділу	26
4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ	28
4.1 Засоби розробки	28
4.2 Вимоги до технічного забезпечення	29
4.2.1 Загальні вимоги	29
4.3 Архітектура програмного забезпечення	30
4.3.1 Діаграма класів	31
4.3.2 Діаграма послідовності	33
4.3.3 Діаграма компонентів	35
4.3.4 Специфікація функцій	36
Висновок до розділу	38
5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ	39

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		4

5.1	Керівництво користувача.....	39
5.2	Випробування програмного продукту.....	48
5.2.1	Мета випробувань.....	48
5.2.2	Загальні положення.....	48
5.2.3	Результати випробувань.....	48
	ЗАГАЛЬНІ ВИСНОВКИ.....	52
	ПЕРЕЛІК ПОСИЛАНЬ.....	53
	Додаток А.....	54

ВСТУП

Дипломний проект присвячений розробці підсистеми продажу та обліку фармацевтичних препаратів ERP системи у вигляді веб-застосування. Призначення цього проекту полягає в тому, щоб полегшити взаємодію між працівниками та клієнтами аптечного закладу, а також надати можливість ведення аптечного закладу в електронній системі здоров'я України e-Health.

Основною особливістю веб-застосування є можливість ведення електронних рецептів.

Веб-застосування для діяльності аптечного закладу є лише складовою великої системи, яка складається з Медичних Інформаційних Систем, які в свою чергу створюють центральний компонент.

Актуальність теми – люди постійно хворіють і потребують лікування. Програмне забезпечення значно полегшить облік купівлі та продажу фармацевтичних препаратів.

Мета – створення підсистеми продажу та обліку фармацевтичних препаратів ERP системи у вигляді веб-застосування.

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		6

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Опис предметного середовища

Багато компаній, підприємств, по мірі свого зростання потребують певні програмні забезпечення, системи для автоматизації їхньої роботи в загальному. Це стосується в більшості великий підприємств, оскільки в малому бізнесі можуть обійтись без цих інструментів.

ERP система - (англ. Enterprise Resource Planning System — Система планування ресурсів підприємства) - корпоративна інформаційна система (KIC), призначена для автоматизації обліку й керування.

Насамперед це система, яка дозволяє зберігати та обробляти більшість критично важливих даних для компанії.

Усі ERP-системи незалежно від їхнього класу об'єднує одна спільна архітектура: платформа (основні та базові можливості, наприклад API (англ. application programming interface), управління даними, базовий функціонал, програмний код); модулі (для внутрішніх користувачів, для зовнішніх користувачів).

Внаслідок медичної реформи в Україні, яка запровадила перенесення усіх даних пацієнтів в електронну систему здоров'я e-Health, з'явилася потреба інтегруватись з нею та взаємодіяти на рівні ERP систем.

У 2016 році реформа охорони здоров'я була в трійці першочергових реформ згідно з опитуваннями населення, поряд із антикорупційною та судовою системами. Восени 2016 року новий уряд прийняв рішення про створення системної реформи в галузі охорони здоров'я.

Предметним середовищем дипломного проекту є веб-застосування аптеки.

Основним призначенням застосування є облік продажів фармацевтичних препаратів та погашення електронних рецептів, взаємодія з електронною системою здоров'я e-Health. Другорядні призначення: ведення звітності, прийом фармацевтів на роботу, їх звільнення.

					ДП ІС-6223.00.000.ПЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

1.1.1 Опис процесу діяльності

Розглянемо дії, які має виконувати фармацевт в межах веб-застосування для аптеки.

1.1.2 Опис функціональної моделі

Для проектування схеми системи необхідно визначити дійових осіб, які приймають участь у процесі (акторів) та визначити їхні ролі.

Акторами в системі є директор аптеки та фармацевт.

Директор аптеки – головний співробітник аптеки, який займається керуванням даних на веб-ресурсі, редагуванням даних інших співробітників (фармацевтів), а також веденням звітності.

Фармацевт – співробітник аптеки, який займається погашенням електронних рецептів, які отримують від пацієнтів, що приходять до аптеки; продажем ліків.

Варіанти використання розподілені по групам користувачів:

Директор аптечного закладу:

- ведення аптечного закладу в електронній системі e-Health;
- ведення працівників аптечного закладу;
- отримання звітів.

Фармацевт:

- ведення власного облікового профілю;
- перегляд електронних рецептів.
- ведення електронних рецептів;
- погашення електронних рецептів;
- перегляд інформативно сторінки рецепту.

Схема структурна варіантів використання представлена на рисунку 1.3 та наведена у графічному додатку.

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		8

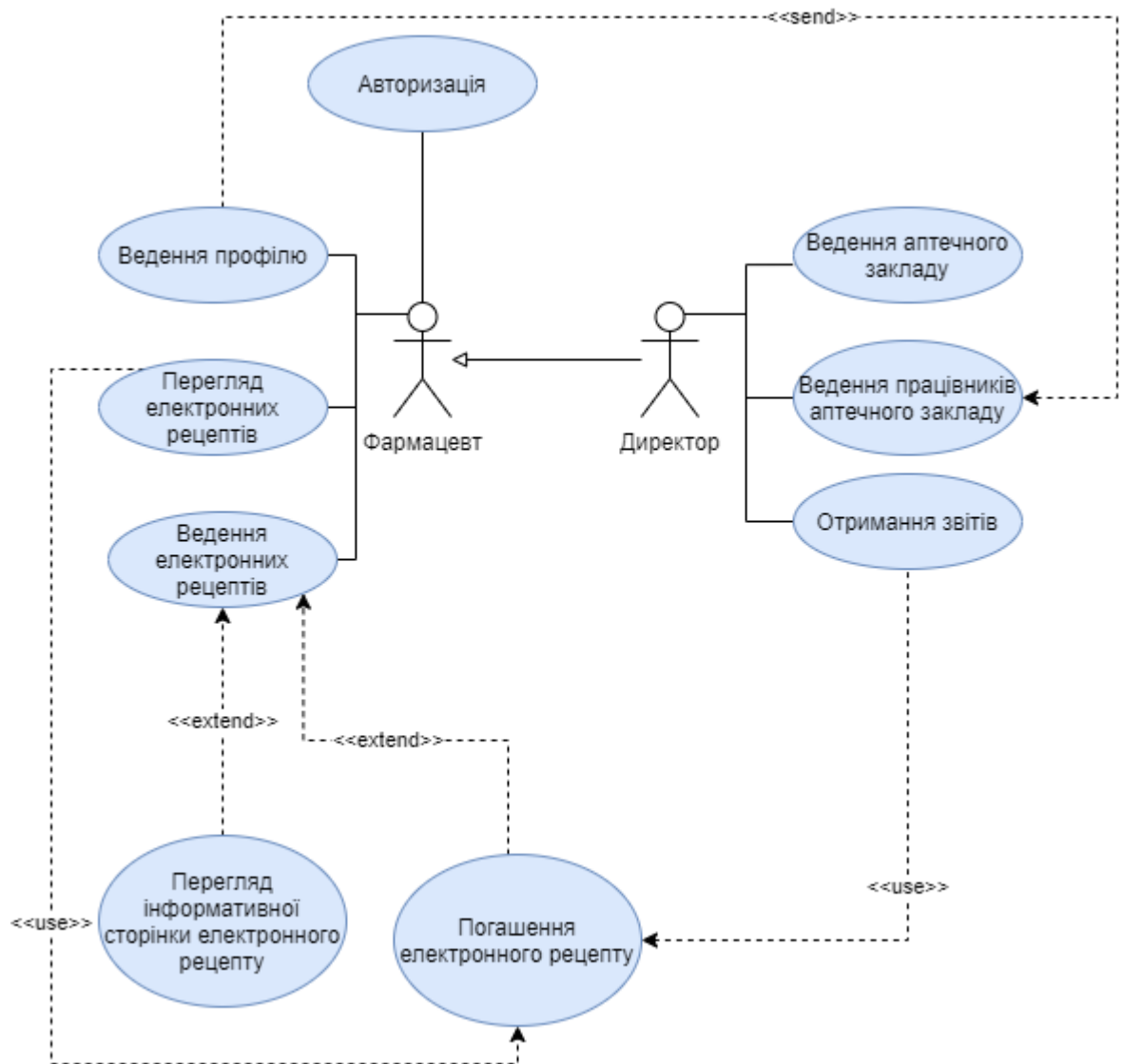


Рисунок 1.3 – Схема структурна варіантів використання

Кожен з варіантів використання має власні вимоги, пріоритети та актора.

Функціональні вимоги до варіантів використання наведені у таблиці 1.1.

Таблиця 1.1 – Вимоги до варіантів використання

Актор	Варіант використання	Опис вимог варіанта використання	Пріоритети
Директор аптеки	Реєстрація аптеки в електронній системі e-Health	RQ 001. Користувач отримує сторінку з формою реєстрації аптечного закладу.	Високий
	Редагування аптеки в електронній системі e-Health	RQ 002. Користувач отримує сторінку з формою реєстрації аптечного закладу.	Високий
	Відновлення паролю	RQ 003. На електронну пошту директора відправляється електронний лист, у якому є посилання, яке дає змогу змінити пароль, якщо перейти за ним. RQ 003.1. Посилання, яке відправляється на пошту може бути використане один.	Високий Середній
	Зміна паролю	RQ 004. Сторінка з формою з полями старого паролю (для підтвердження користувача), нового паролю та підтвердженням нового паролю. RQ 004.2. Поля нового паролю та підтвердження нового паролю співпадають.	Високий Високий

Продовження таблиці 1.1

	Авторизація	RQ 005. Користувач заповняє поля логіну та паролю для авторизованого входу в систему.	Високий
	Додавання фармацевтів в аптечному закладі	RQ 006. Користувач отримує сторінку з формою реєстрації фармацевта.	Високий
	Редагування фармацевтів в аптечному закладі	RQ 007. Користувач отримує сторінку з формою редагування фармацевта.	Середній
	Звільнення фармацевтів в аптечному закладі	RQ 008. Користувач отримує сторінку з формою звільнення фармацевта.	Високий
	Формування звітів	RQ 009. Користувач отримує сторінку з полями для введення періоду звітності та кнопкою, що дає змогу сформувати звіт.	Середній
Фармацевт	Відновлення паролю	RQ 0010. На електронну пошту директора відправляється електронний лист, у якому є посилання, яке дає змогу змінити пароль RQ 0010.1. Посилання, яке відправляється на пошту може бути використане один.	Високий Середній

Продовження таблиці 1.1

	Зміна паролю	RQ 0011. Сторінка з формою з полями старого паролю (для підтвердження користувача), нового паролю та підтвердженням нового паролю. RQ 0011.2. Поля нового паролю та підтвердження нового паролю співпадають.	Високий Високий
	Авторизація	RQ 0012. Користувач заповняє поля логіну та паролю для авторизованого входу в систему.	Високий
	Пошук електронного рецепту	RQ 0013. Користувач отримує сторінку з полем введення номера електронного рецепту.	Високий
	Редагування та попередній перегляд електронного рецепту	RQ 0014. Після успішного знаходження електронного рецепту в базі даних електронної системи здоров'я e-Health, користувач отримує сторінку з повною інформацією про рецепт, а також функціоналом, який дозволяє редагувати його.	Високий
	Продаж ліків	RQ 0015. Користувач отримує сторінку про підтвердження погашення рецепту та вартістю включених в нього ліків.	Високий

1.2 Огляд наявних аналогів

Багато аптек власними силами намагаються налагодити внутрішні бізнес-процеси за рахунок створення власних програмних забезпечень, які надають базовий мінімальний функціонал. Всі ці програми мають як свої позитивні риси так і негативні. В основному їхні мінуси полягають в тому, що ці програми працюють в тестових режимах, надають мінімум функціоналу та збільшують ризики неправильного обліку продажів.

Суть мого проекту полягає в тому, щоб повністю забезпечити аптеку інтеграцією з електронною системою e-Health, автоматизувати більшість процесів всередині аптеки, надати можливість слідкувати за продажами за рахунок ведення звітності. Веб-застосування відрізняється від аналогів тим, що ввібрав в себе все найкраще від них, відкинув все зайве, та написаний на нових технологіях.

Тому основними особливостями проектованої системи є:

- архітектура непрямої взаємодії клієнтської сторони з сервером і безпосередньо базою даних.
- гнучкість системи в архітектурному контексті:
- підключення нових модулів не погіршує роботу системи, а навпаки, дозволяє підтримувати актуальність системи і підвищувати її продуктивність;
- структура бази даних дозволяє використовувати її будь-якій аптеці.
- «user friendly» дизайн, тобто дизайн, інтуїтивно зрозумілий для будь-якого користувача і не знайдений ні в одному з аналогів цієї системи. Це дуже важливий аспект системи, тому що мало хто хоче працювати в незрозумілому інтерфейсі, плюс зрозуміле веб-застосування прискорює роботу аптеки в цілому.
- дизайн інтерфейсу адаптується до різних розмірів екрану (планшети, мобільні телефони, ноутбуки, монітори). Цей показник не є поширеним серед ERP-систем в цілому, тому дана система має очевидну перевагу серед інших аналогів.

Як аналоги розглядались наступні веб-ресурси:

- <https://www.infomed.ck.ua/>;

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		13

- <https://medcore.com.ua/>;
- <https://pharmbase.com.ua/uk/>.

Посилання бралися з офіційного ресурсу системи e-Health <https://ehealth.gov.ua/aptechni-zaklady>.

Вважаю, що наведені переваги значно перебільшують інші переваги аналогів та цього достатньо, щоб конкурувати з ними.

					ДП ІС-6223.00.000.ПЗ	Арк.
						14
Змн.	Арк.	№ докум.	Підпис	Дата		

1.3 Постановка задачі

1.3.1 Призначення розробки

Призначенням є контролювання продажів та обліку фармацевтичних препаратів, взаємодії працівників та клієнтів аптечних закладів.

1.3.2 Цілі та задачі розробки

Цілі створення веб-застосування аптечного закладу:

- полегшити взаємодію працівників аптечного закладу та клієнтів;
- спростити процес продажу фармацевтичних препаратів.

Для реалізації поставлених цілей система має вирішувати наступні задачі:

- ведення аптечного закладу;
- ведення працівників аптечного закладу;
- ведення електронних рецептів;
- отримання звітності.

Висновок до розділу

В ході написання розділу було проаналізовано предметна область, виявлені існуючі аналоги і сформульована постановка задачі.

У розділі описується предметне середовище аптеки не тільки в контексті розробки веб-застосування, але і в цілому для розуміння завдань по правильній роботі веб-застосування аптечного закладу, описується функціональна модель і функціональні вимоги до системи.

У цьому розділі описуються цілі розробки та завдання по їх досягненню, щоб веб-застосування працювало правильно і відповідало вимогам, викладеним в функціональній моделі. У розділі також перераховані цілі системи.

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		16

2 ІНФОРМАЦІЙНЕ ЗАБЕЗПЕЧЕННЯ

2.1 Вхідні дані

Вхідні дані до веб-застосування аптечного закладу надходять з декількох джерел. Такими джерелами є користувачі різних типів (директор аптечного закладу, фармацевт).

Від директора надходять реєстраційні дані аптечного закладу.

Від фармацевта надходить інформація про редагування електронного рецепту та продажі фармацевтичних препаратів.

2.2 Вихідні дані

Вихідними даними є інформація про погашений рецепт, його вартість та склад, звіти продажів фармацевтичних препаратів на сторінці директора аптечного закладу.

2.2.1 Опис структури бази даних

Наведено опис таблиць розробленої бази даних у таблицях 2.1-2.8.

Таблиця 2.1 – Таблиця конфігураційних параметрів.

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
Configurations	Id	Integer	Ідентифікатор
	Name	varchar(700)	Назва параметру
	Value	varchar(700)	Значення параметру
	Description	varchar(700)	Опис параметру

Таблиця 2.2 – Таблиця міграцій бази даних.

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
__RunedScripts	Id	Integer	Ідентифікатор
	Name	varchar(700)	Назва міграції
	Script	varchar(MAX)	Скрипт
	RunedDate	datetime	Дата виконання

Таблиця 2.3 – Таблиця ролей користувачів.

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
_DICT_Roles	Id	Integer	Ідентифікатор
	Name	varchar(70)	Назва параметру
	Created	datetime	Дата створення
	Updated	datetime	Дата оновлення

Таблиця 2.4 – Таблиця користувачів.

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
Users	Id	Integer	Ідентифікатор
	roleId	Integer	Ідентифікатор ролі
	login	varchar(60)	Логін користувача
	password	varchar(60)	Хеш паролю користувача
	code	varchar(60)	Код користувача
	IsLogged	bit	Булеве значення залогіненого
	LastAction	varchar(60)	Остання дія користувача
	Created	datetime	Дата створення
	Updated	datetime	Дата оновлення

Таблиця 2.5 – Таблиця токенів користувачів.

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
UserToken	Id	Integer	Ідентифікатор
	UserId	varchar(70)	Назва параметру
	token	varchar(200)	Токен користувача
	Created	datetime	Дата створення
	Updated	datetime	Дата оновлення

Таблиця 2.6 – Таблиця фармацевтів.

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
Pharmacists	Id	Integer	Ідентифікатор
	FirstName	varchar(150)	Ім'я
	LastName	varchar(150)	Прізвище
	MiddleName	varchar(150)	По-батькові
	Created	datetime	Дата створення
	Updated	datetime	Дата оновлення
	UserId	Integer	Ідентифікатор користувача
	Adress	varchar(150)	Адреса
	ImageUri	varchar(150)	Посилання на фото профілю
	BirthDate	datetime	Дата народження
	Email	varchar(150)	Електронна пошта
	Phone	varchar(150)	Мобільний телефон
	GenderId	Integer	Ідентифікатор статі

Таблиця 2.7 – Таблиця гендеру.

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
_DICT_Genders	Id	Integer	Ідентифікатор
	Name	varchar(70)	Назва параметру
	Name_en	varchar(70)	Назва параметру англійською
	Created	datetime	Дата створення
	Updated	datetime	Дата оновлення

Таблиця 2.8 – Таблиця об'єктів, що взаємодіють з електронною системою e-Health.

Назва таблиці	Назва стовпця	Тип даних	Детальна інформація
EhealthObjectProfile	Id	Integer	Ідентифікатор
	EntryId	varchar(70)	Ідентифікатор сутності
	EntryType	varchar(70)	Тип Сутності
	Created	datetime	Дата створення
	Updated	datetime	Дата оновлення
	Status	varchar(70)	Статус
	JsonData	varchar(MAX)	Дані у форматі json
	EhealthId	varchar(70)	Ідентифікатор об'єкту у e-Health

Схема бази даних зображена на рисунку 2.1 та наведена у графічному додатку.

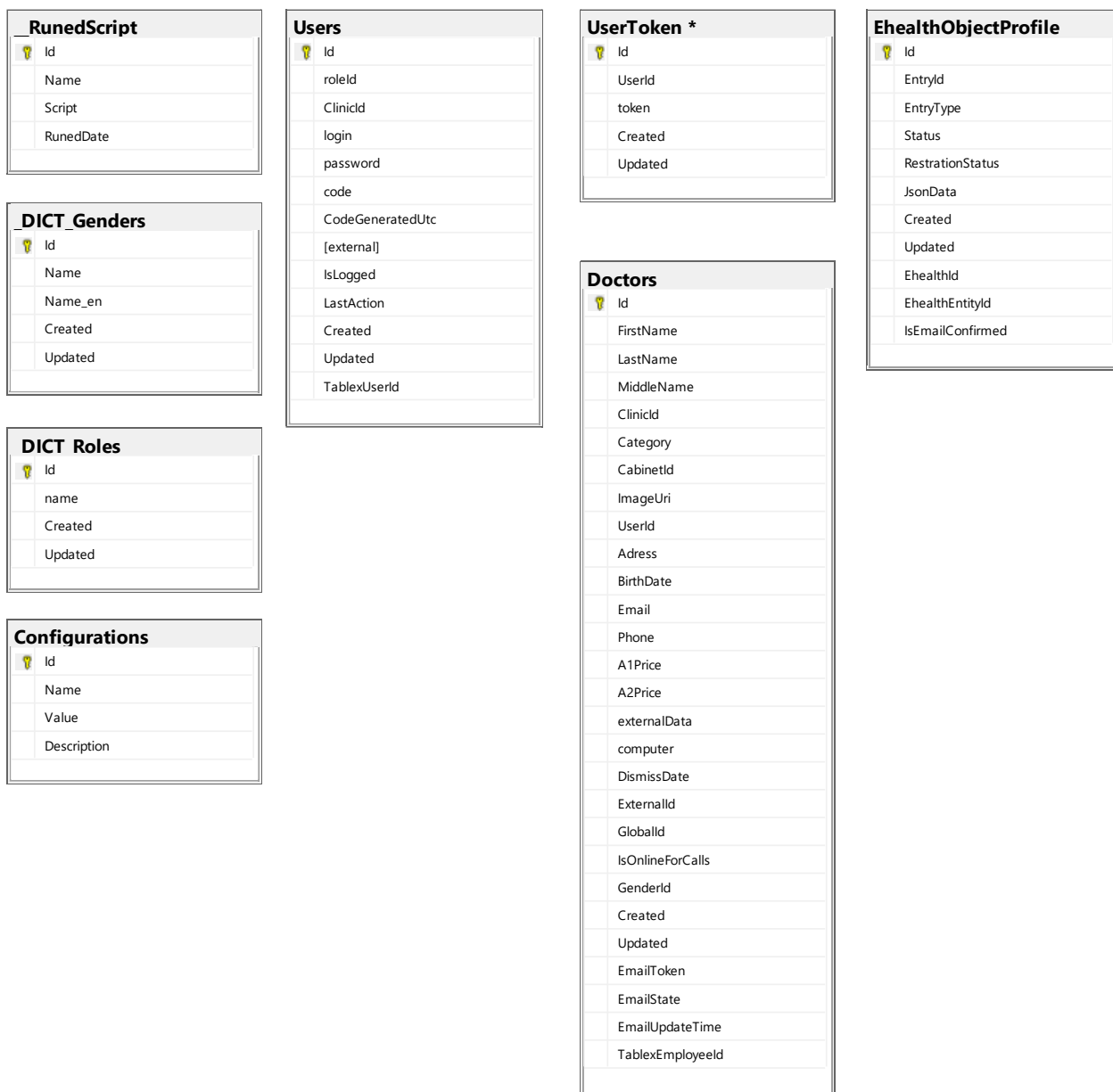


Рисунок 2.1 - Схема бази даних

Висновок до розділу

У розділі описуються вхідні дані веб-застосування аптечного закладу, джерела вхідних даних. Описує вихідні дані, структуру вихідних даних, що представляють таблиці.

Вхідні дані описують джерела вхідних даних і яку інформацію надає кожен з джерел.

Також надається список вихідних даних для веб-застосування аптеки і описується, що вони з себе представляють.

У розділі докладно описується структура бази даних, представлена структура кожної з таблиць. Він показує, як кожна таблиця представлена в самій структурі бази даних, і які типи даних використовуються для кожного з стовпців. Крім того, кожен стовпець має семантичний опис для кращого розуміння роботи конкретної таблиці.

У графічному додатку наведена схема бази даних, що показує зв'язки між таблицями. Показує відносини між таблицями, зовнішніми і внутрішніми ключами. Показані типи внутрішніх ключів (звичайні і складові) і зовнішні ключі (один до багатьох, багато один до одного, багато до багатьох).

Кількість таблиць і їх структура дозволяє зберігати всі необхідні дані і розширювати систему.

3 МАТЕМАТИЧНЕ ЗАБЕЗПЕЧЕННЯ

3.1 Змістовна постановка задачі

Можливість створювати електронні рецепти і викупувати їх - це нововведення, тому корисно знати, як це нововведення впливає на кількість користувачів, які купують ліки по електронним рецептами, оскільки воно є основним показником ефективності роботи веб-застосування аптеки. Кількість проданих ліків буде основним показником, оскільки у пацієнтів, які купують ліки за рецептом, вибір обмежений. Тому завдання полягає в тому, щоб знайти зміни в кількості продажів ліків до введення електронних рецептів та після.

3.2 Математична постановка задачі

Завдання полягає в тому, щоб знайти зміни в кількості продажів ліків до і після впровадження електронної рецептурної системи.

Вхідні дані: кількість фармацевтичних препаратів, проданих до і після інновації.

3.3 Обґрунтування методу розв'язання

Аналіз результатів покаже, чи були інновації в формі електронних рецептів ефективними, і спланує нові інновації для підвищення ефективності продажів лікарських засобів. В якості методу вирішення проблеми був обраний метод порівняння двох тенденцій: до інновацій і після.

Тренд - основний тренд часового ряду. Тенденції можуть бути описані різними рівняннями - лінійними, логарифмічними, експонентними [4]. Фактичний тип тренда визначається на основі вибору його функціональної моделі статистичними методами або згладжування початкових часових рядів.

Лінії тренда - це елемент апарату технічного аналізу, який використовується, зокрема, для виявлення трендів цін на різних типах бірж.

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		23

3.4 Опис методів розв'язання

Позначимо через w_{ij} кількість продажів ліків аптечного закладу до нововведень j -го дня, i -го тижня. Результати зазвичай подаються у вигляді матриці $\{w_{ij}\}$ з t рядками і семи стовпцями ($i = 1, \dots, t; j = 1, \dots, 7$).

Позначимо через v_{ij} кількість продажів ліків аптечного закладу після нововведень j -го дня, i -го тижня. Результати тестування зазвичай представляються у вигляді матриці $\{v_{ij}\}$ з n рядками і семи стовпцями ($i = 1, \dots, t; j = 1, \dots, 7$).

Процес статистичної обробки матриці продажів ліків аптечного закладу будемо розглядати крок за кроком.

1. Обчислюється середня кількість проданих ліків до нововведень за тиждень w_i ($i = 1, \dots, t$), де i – номер тижня:

$$w_i = \frac{\sum_{j=1}^d w_{ij}}{d} \quad (i = 1, \dots, t; j = 1, \dots, d)$$

2. Обчислюється середня кількість проданих ліків після нововведень за тиждень v_i ($i = 1, \dots, t$), де i – номер тижня:

$$v_i = \frac{\sum_{j=1}^d v_{ij}}{d} \quad (i = 1, \dots, t; j = 1, \dots, d)$$

3. На основі отриманих числових рядів w_i та v_i будуємо тренди для кожного з них. Щоб бачити наскільки ефективними були зміни.

Вхідні дані по тижням до і після нововведень зображені на таблиці 3.1.

Таблиця 3.1 – Таблиця з вхідними даними з продажів ліків за кожен тиждень до і після нововведень

i	w_i	v_i
1	2676	2629
2	2737	2655
3	2884	2819
4	2608	2757
5	2963	3077

Продовження таблиці 3.1

6	2858	2961
7	2776	2928
8	2646	2776
9	2734	2693
10	3126	3056
11	2562	2737
12	2979	2947
13	2816	2723
14	3151	3269
15	3177	3264
16	3180	3271
17	2462	2576
18	2944	2953
19	2317	2387
20	2530	2576

На рисунках 3.1 та 3.2 зображено тренди з прогнозами на майбутні двадцять тижнів продажів ліків до і після інновацій відповідно. Тренди є лінійними, тобто у наступному форматі $y = cx * d$.

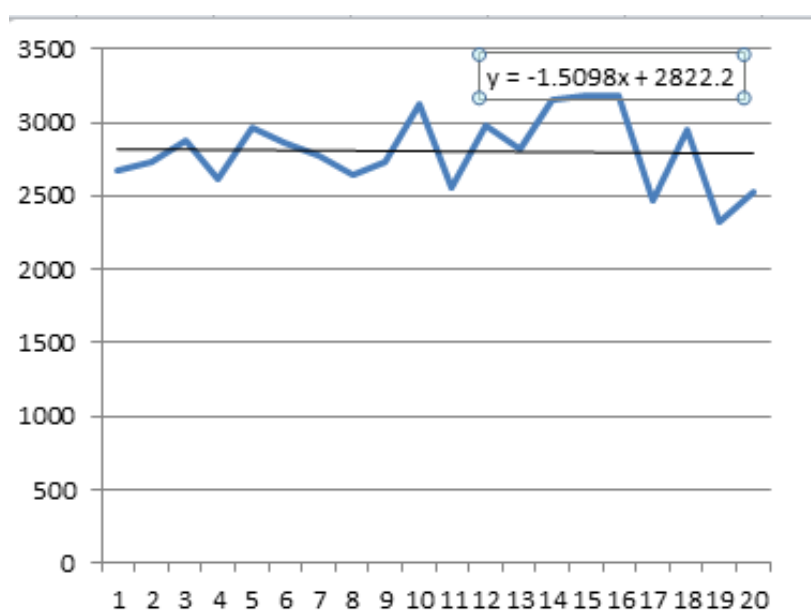


Рисунок 3.1 – Тренд продажів ліків до новацій

Змн.	Арк.	№ докум.	Підпис	Дата

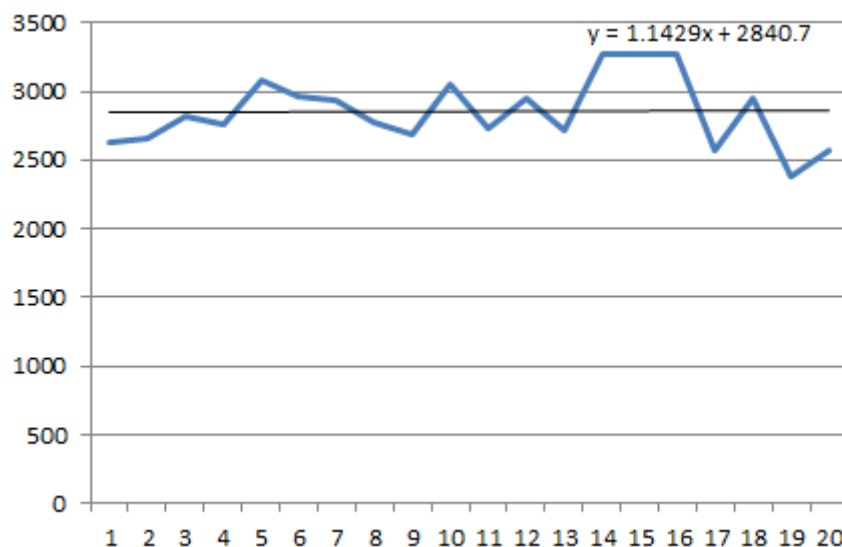


Рисунок 3.2 – Тренд продажів ліків після новацій

Як видно з графіків $c_{\text{до}} = -1.5098$ та $c_{\text{після}} = 1.1429$, тобто $c_{\text{до}} < c_{\text{після}}$.

До новацій тренд продажів має від'ємний коефіцієнт, що говорить про його падіння, після нововведення – додатний коефіцієнт.

Це означає, що ефективність продажів ліків зросла, нововведення були виправдані.

Змн.	Арк.	№ докум.	Підпис	Дата

Висновок до розділу

У цьому розділі була сформульована змістовна та математична постановка задачі. Завдання пошуку зміни ефективності веб-застосування аптеки після надання можливості продавати ліки за електронним рецептом була вирішена.

При вирішенні цієї проблеми вхідними даними було кількість продажів ліків за кожен тиждень після нововведень і до нововведень.

При написанні цього розділу використовувався математичний статистичний апарат та інструменти для побудови графіків і трендів в MS Excel 2015.

В результаті ми приходимо до висновку, що завдяки можливості продавати рецептурні ліки ефективність продажу ліків зросла приблизно в 2 рази, і це є хорошим показником ефективності інновацій в системі електронних рецептів.

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		27

4 ПРОГРАМНЕ ТА ТЕХНІЧНЕ ЗАБЕЗПЕЧЕННЯ

4.1 Засоби розробки

При створенні програмного продукту використовувались мова програмування C#, ASP.NET MVC 4, EF 6, система управління базами даних Microsoft. Платформа AngularJS також використовується на стороні клієнта.

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		28

4.2 Вимоги до технічного забезпечення

4.2.1 Загальні вимоги

Для правильної роботи цієї програми технічні засоби повинні включати: процесор з високою тактовою частотою та достатнім об'ємом оперативної пам'яті.

					ДП ІС-6223.00.000.ПЗ	Арк.
						29
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3 Архітектура програмного забезпечення

У цьому дипломному проекті використовувався наступний архітектурний підхід - розділення на три рівня.

Розглянемо кожен з них:

- клієнтський рівень (Web) відповідає за взаємодію клієнта з програмою через веб-браузер;
- рівень бізнес-логіки (Business logic) відповідає за всю бізнес-логіку програми і функціональність веб-застосування. Цей рівень містить всі критично важливі для бізнесу правила;
- рівень доступу до даних відповідає за доступ до бази даних. Цей шар взаємодіє з базою даних через фреймворк Entity.

					ДП ІС-6223.00.000.ПЗ	Арк.
						30
Змн.	Арк.	№ докум.	Підпис	Дата		

4.3.1 Діаграма класів

Діаграма класів повністю повторює структуру бази даних, тобто все таблиці, які в ній знаходяться. Ця структура зумовлена використанням бібліотеки Entity Framework, в цьому випадку використовується перший підхід до бази даних. Для кожної таблиці в базі даних створюється клас, який копіює структуру таблиці і всі її типи даних, тому немає сенсу розглядати класи рівня бази даних.

На діаграмі класів розглянемо класи рівня обслуговування, так як вони є основою бізнес-логіки і найкращим чином демонструють роботу системи. Схема структурна класів зображена на рисунку 4.1 та у графічному додатку.



Рисунок 4.1 – Схема структурна класів сервісного рівня

4.3.2 Діаграма послідовності

На схемі структурній послідовності розглянемо процес погашення електронного рецепту через класи, які зображені на рисунку 4.1.

Таблиця 4.1 – Об'єкти діаграми послідовності

Клас	Відповідальність
EhealthWebService	Клас, який відповідає за передачу запитів до електронної системи e-Health.
AppServiceEhealth	Клас відповідальний за формування запитів до електронної системи e-Health.
AppServiceAuth	Клас відповідальний за авторизацію користувачів.
AppServicePharmacist	Клас відповідальний за роботу з фармацевтами.

Схема структурна послідовності погашення електронних рецептів зображена на рисунку 4.2 та наведена у графічному додатку.

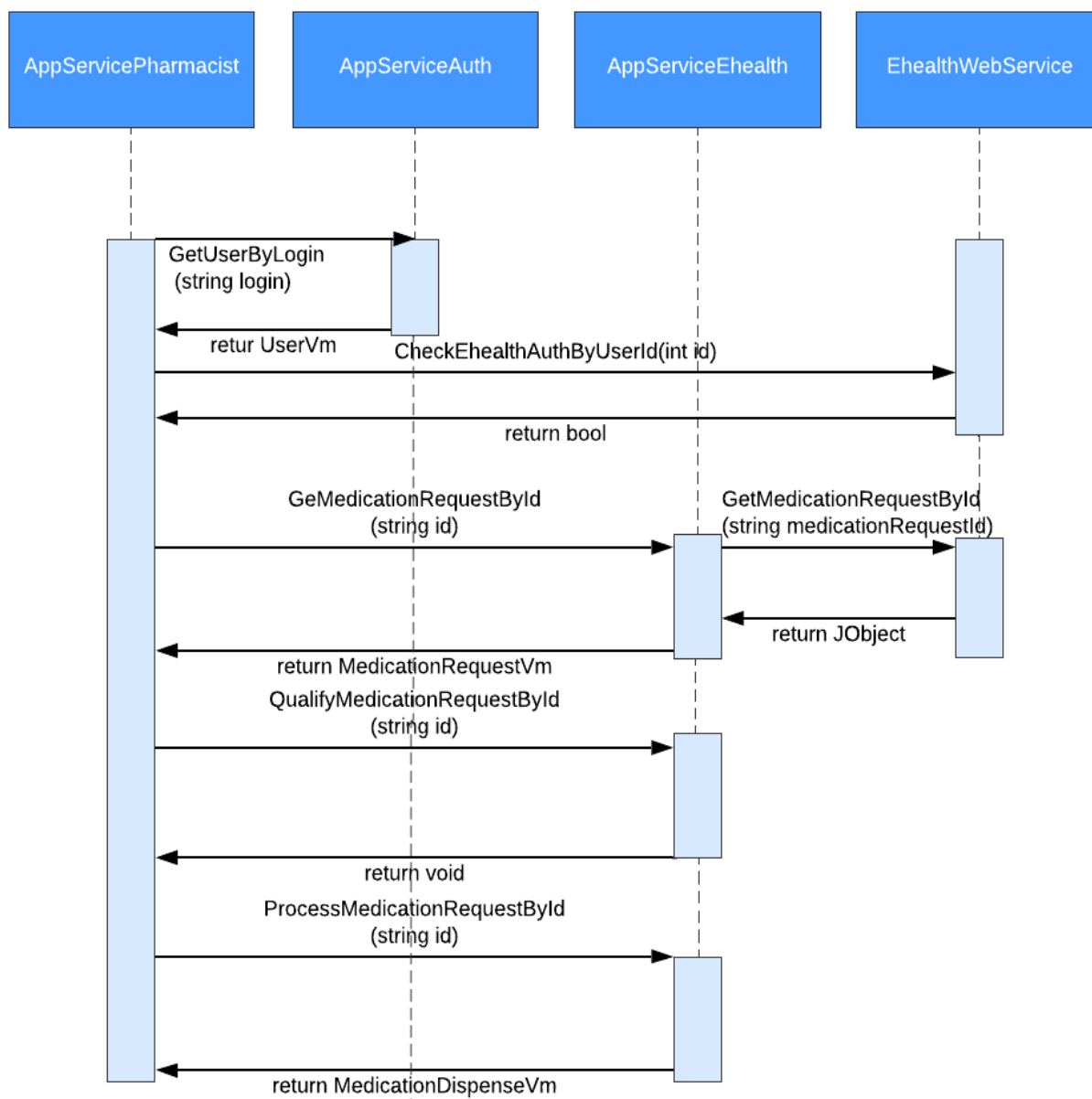


Рисунок 4.2 – Схема структурна послідовності

4.3.3 Діаграма компонентів

На цій діаграмі задіяні об'єкти, які представлені на таблиці 4.2.

Таблиця 4.2 – Об'єкти діаграми компонентів

Тип файлу	Опис типу файлу
html file	Містить загальну структуру сторінок сайту, інформацію та елементи, які знаходяться на них.
css file	Містить стилі сайту та положення, як елементи мають знаходитись на сторінці.
ts file	Клієнтська функціональна складова сайту, всі клієнтські функції та механізми, які на ньому відбуваються описані тут.
Database	Містить дані сайту, які виникають при взаємодії з ним.
cs file	Містить дані про серверні функції та логіку серверної сторони сайту.

Схема структурна компонентів зображена на структурній схемі компонентів на рисунку 4.3 та наведена у графічному додатку.

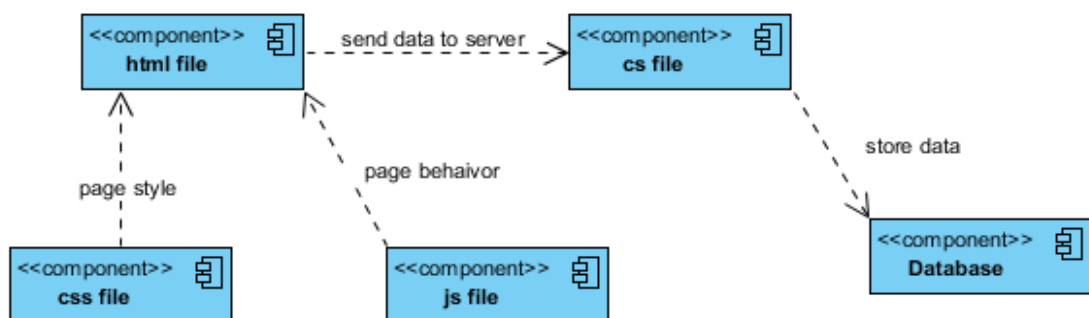


Рисунок 4.3 – Схема структурна компонентів

4.3.4 Специфікація функцій

Функції класів частини програмного забезпечення наведені в таблиці 4.3.

Таблиця 4.3 – Функції класів програмного забезпечення

Назва	Примітка
public bool CheckEhealthLoginByUserId(int id)	Перевіряє наявність авторизації в електронній системі e-Health.
public void CreateUser(UserVm user)	Створює користувача в системі.
public UserVm GetUser(int id)	Отримує інформацію про користувача по ідентифікатору в системі.
public void ResetPassword(string oldPassword, string newPassword, string onfirmPassword)	Змінює пароль.
public void ForgotPassword(string login, string newPassword)	Відправляє на поштову адресу користувача лист для відновлення паролю.
public void CreateOrUpdateLegalEntity(LegalEntityVm legalEntity)	Створює або оновлює дані аптечного закладу.
public void ConfirmLegalEntity(LegalEntityVm legalEntity)	Підтверджує зміни даних аптечного закладу.
public MedicationRequestVm GetMedicationRequestById(string id)	Отримує електронний рецепт за ідентифікатором.
public void QualifyMedicationRequestById(string id)	Перевіряє актуальність електронного рецепту та отримує список ліків.

Продовження таблиці 4.3

public MedicationDispenseVm ProcessMedicationRequestById(string id)	Обробляє запит на погашення електронного рецепту. Повертає модель з погашенням.
public void Login(LoginDto login)	Логінить користувача в систему.
public void Logout(string login)	Вилогінює користувача із системи.
public LegalEntityVm GetLegalEntityDetailsById(int id)	Отримує детальну інформацію про аптечний заклад.
public PharmacistVm GetPharmacistById(int id)	Отримує інформацію про співробітника аптеки.
public List< PharmacistVm > GetPharmacistList ()	Отримує список співробітників аптеки.

Змн.	Арк.	№ докум.	Підпис	Дата

Висновок до розділу

У розділі докладно описуються засоби розробки, які використовуються в процесі створення веб-застосування аптечного закладу, і перераховуються основні переваги кожного з них. Також було дано докладний опис кожного програмного забезпечення, використаного при розробці веб-застосування аптечного закладу, і точно описано, як працюють всі його компоненти і які підходи слід використовувати при роботі з ними.

Описані вимоги до технічного забезпечення, які потрібні для використання комплексу задач. Вимоги до обладнання включають вимоги до сервера, на якому буде працювати веб-додаток, і клієнтської частини, яка повинна бути встановлена на стороні клієнта, щоб веб-додаток функціонував належним чином.

У цьому розділі докладно описується архітектура сайту і обґрунтовується вибір такого архітектурного підходу на кожному рівні, а також пояснюється, чому ця архітектура є найкращою і які переваги має веб-застосування.

У цьому розділі представлені схеми структурні класів, послідовностей і компонентів, які описують архітектуру системи і її частин, а також описуються деякі функції рівня обслуговування для кращого розуміння завдань, які виконуються системою.

Вимоги, викладені в технічній підтримці, накладають обмеження на використання продукту. Якщо середовище не відповідає вимогам, продукт не запуститься.

5 ТЕХНОЛОГІЧНИЙ РОЗДІЛ

5.1 Керівництво користувача

Для входу на сервіс необхідно набрати у адресному рядку будь-якого сумісного браузера адресу сайту і натиснути «Перейти».

Після цього буде відображена головна сторінка сайту, на якій міститься форма авторизації. Для повноцінної роботи на веб-сайті необхідно увійти в систему під власним обліковим записом. Для здійснення авторизації необхідно увійти в систему за допомогою локального профілю. Вигляд сторінки авторизації та одночасно головної сторінки показано на рисунку 5.1.

Рисунок 5.1 – Головна сторінка

Для успішної авторизації введений логін користувача повинен існувати у системі та пароль повинен бути введеним коректно та бути активними.

У випадку втрати паролю користувачем існує функція відновлення паролю. Для цього необхідно зі сторінки логіну перейти на сторінку відновлення паролю та ввести номер мобільного телефону. Система перевіряє наявність цього номеру в системі та у разі успішної перевірки відправляє SMS-код на відновлення паролю. Вигляд сторінки відновлення паролю зображено на рисунку 5.2.

Рисунок 5.2 – Сторінка відновлення паролю

Після вводу коду отриманого в SMS-повідомленні, користувач потрапляє на сторінку з вводом нового паролю. На цій сторінці необхідно ввести новий пароль та підтвердити його. Вигляд сторінки вводу нового паролю зображено на рисунку 5.3.

Рисунок 5.3 – Сторінка вводу нового паролю

Після авторизації директора аптеки відображається сторінка з інформацією про аптечний заклад. Вигляд сторінки з інформацією про аптечний заклад зображено на рисунку 5.4.

Рисунок 5.4 – Сторінка з інформацією про аптечний заклад

Директор аптеки може переглядати співробітників аптечного закладу, тобто фармацевтів. Для цього він повинен перейти на сторінку з фармацевтами. Вигляд сторінки з фармацевтами зображено на рисунку 5.5.

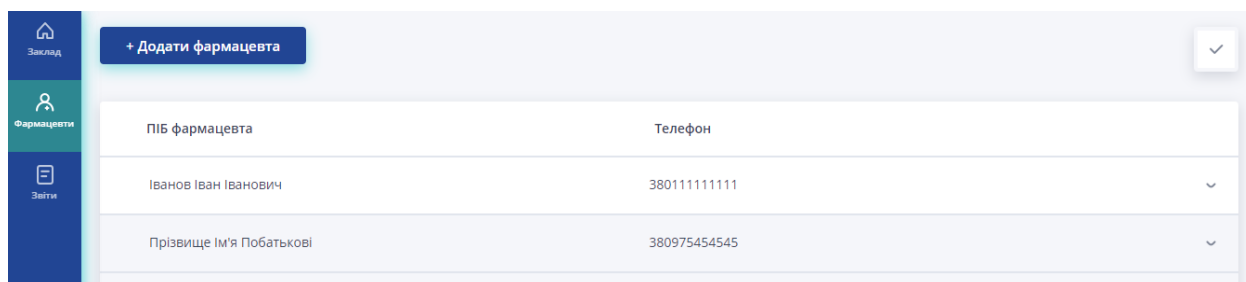


Рисунок 5.5 – Сторінка з фармацевтами

Директор аптеки може реєструвати співробітників аптечного закладу, тобто фармацевтів. Для цього він повинен натиснути кнопку «Додати фармацевта» на сторінці з фармацевтами. Відкриється сторінка з формою реєстрації фармацевта. Вигляд сторінки з формою реєстрації фармацевта зображено на рисунку 5.6.

Створення картки фармацевта

Завантажити фото

* Моб. телефон

+38 (0_) _ _ _ _

* Прізвище

Прізвище

* Ім'я

Ім'я

* По батькові

По батькові

* Стать

☐ Чоловік ☐ Жінка

Дата народження

Дата народження

Ел. адреса

Ел. адреса

* обов'язкові для заповнення поля

✓ Зберегти

✕ Відмінити

Activate Wi-Fi
Go to Settings

Рисунок 5.6 – Сторінка з формою реєстрації фармацевта

У кожного користувача є можливість змінити свій пароль. Для цього треба у мене вибрати пункт «Редагувати», відкриється службова сторінка з опцією «Змінити пароль». Вигляд службової сторінки для редагування зображено на рисунку 5.7.

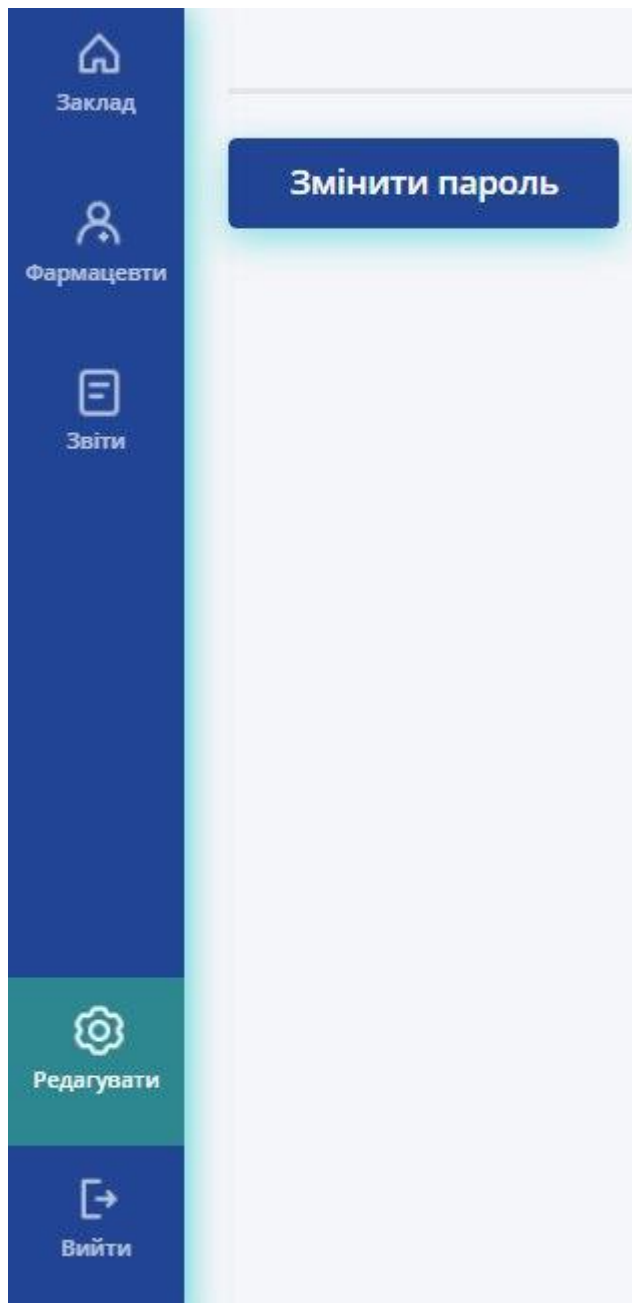


Рисунок 5.7 – Сторінка для редагування профілю

Після того, як користувач натискає кнопку «Змінити пароль» відкривається сторінка з формою для зміни паролю. Вигляд сторінки з формою для зміни паролю зображено на рисунку 5.8.

Рисунок 5.8 – Сторінка з формою для зміни паролю

Директор має можливість сформувати звіт за кількістю виданих ліків. Для цього потрібно перейти на сторінку «Звіти». Вигляд сторінки для формування звіту зображено на рисунку 5.9.

					ДП ІС-6223.00.000.ПЗ	Арк.
						44
Змн.	Арк.	№ докум.	Підпис	Дата		

Звіт про відпущені лікарські засоби та їх вартість

13.05.2020 13.05.2020

Очистити Excel

Рисунок 5.9 – Сторінка для формування звіту

Після того, як у систему авторизується фармацевт, відкривається головна сторінка з полем для пошуку електронного рецепту. Вигляд сторінки з полем для пошуку електронного рецепту зображено на рисунку 5.10.

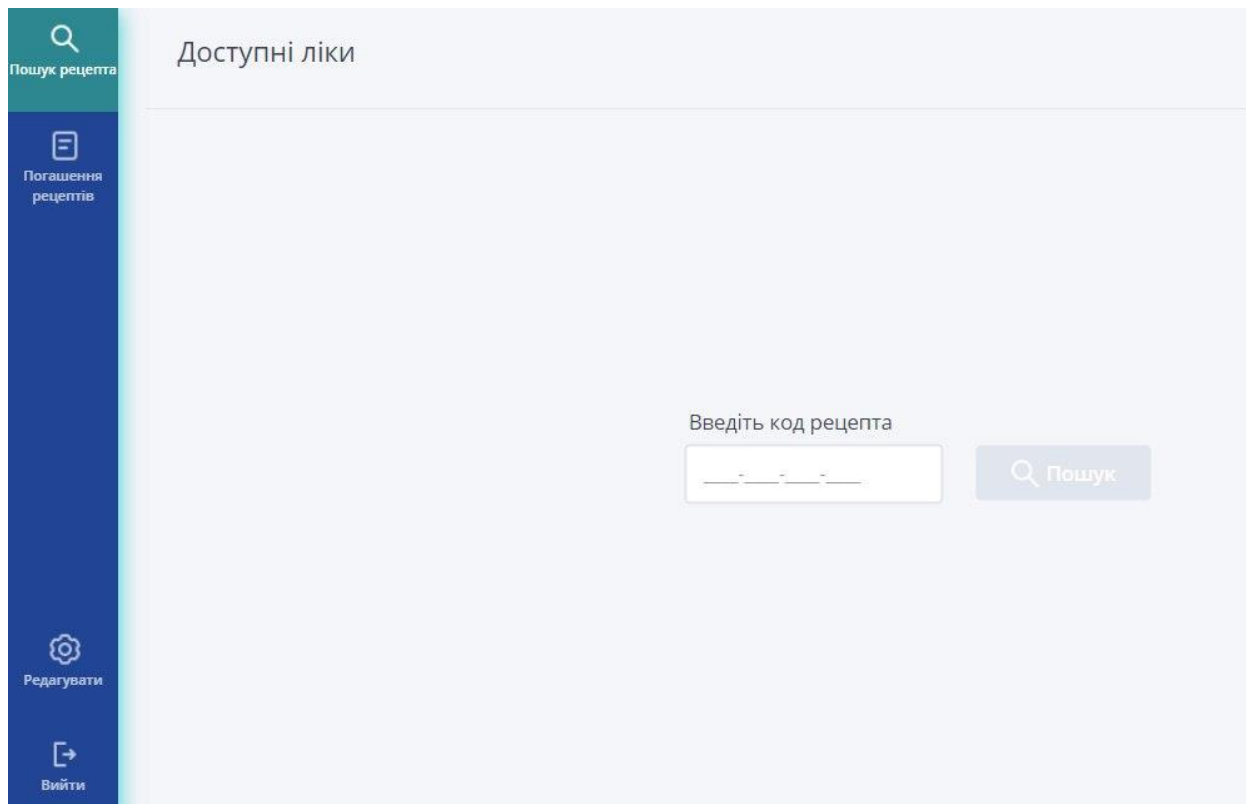


Рисунок 5.10 – Сторінка з полем для пошуку електронного рецепту

Після вдалого пошуку електронного рецепту відкривається сторінка з детальною інформацією про цей рецепт. Вигляд сторінки з детальною інформацією про електронний рецепт зображено на рисунку 5.11.

Пошук рецептів

Погашення рецептів

Редагувати

Статус рецепта

Рецепт 0000-6965-5E28-12AT активний

✓

Міжнародна непатентована назва (МНН)	Дата виписки рецепта	Лікар	Пацієнт
Лозартан 100 MG таблетки №90 таблетка	04.05.2020	Делікатний Фармацевт	Дорошенко В. П.
Медичний заклад	Програма		
Медичний заклад №1	Доступні ліки		

Лікарський засіб	Кількість доз	Роздрібна ціна	Відшкодування за препарат	Доплата пацієнтом	Кількість упаковок
ЛОЗАП® 90 таблетки	90	292,18	156.61	135.57	0

Рисунок 5.11 – Сторінка з детальною інформацією про електронний рецепт

Після успішного погашення рецепту, рецепт додається в список погашень. Цес список можна переглянути, якщо перейти на сторінку «Погашення рецептів». Вигляд сторінки зі списком погашень зображено на рисунку 5.12.

Список погашених рецептів

Параметри пошуку

16.01.2020 13.05.2020 → Опрацьовано

Пошук

Дата створення	Статус	Номер рецепта	Загальна вартість (грн)	Доплата пацієнтом (грн)
16.01.2020	Опрацьовано	0000-M9XX-5K7M-T4X2	80.08	7.00

Рисунок 5.12 – Сторінка зі списком погашень

5.2 Випробування програмного продукту

В цьому підрозділі наведено опис тестів і порядок їх виконання для перевірки на відповідність програмного забезпечення функціональним вимогам, представленим у технічному завданні на створення підсистеми продажу та обліку фармацевтичних препаратів корпоративної ERP системи у вигляді веб-застосування.

5.2.1 Мета випробувань

Метою випробувань являється перевірка відповідності функцій веб-застосування аптечного закладу вимогам висунутим у технічному завданні.

5.2.2 Загальні положення

Випробування проводяться на основі наступних документів:

- ГОСТ 34.603–92. Інформаційна технологія. Види випробувань автоматизованих систем;
- ГОСТ РД 50-34.698-90. Автоматизовані системи вимог до змісту документів.

5.2.3 Результати випробувань

В процесі тестування була перевірена уся функціональність веб-застосування аптечного закладу. У наступній таблиці 5.1 наведений перелік випробувань основних функціональних можливостей та результати випробувань.

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		48

Таблиця 5.1 – Проведення випробувань

№ випроб.	Ситуація (умова)	Ефект (дія)	Ефект фактичний
1	Неавторизований користувач знаходиться на головній сторінці сайту	При переході на адресу веб-застосування, бачить сторінку авторизації	Співпадає з очікуваним
2	Користувач намагається авторизуватися	Користувач вводить коректні логін/пароль, тисне кнопку логіну і потрапляє на головну сторінку	Співпадає з очікуваним
3	Користувач намагається авторизуватися	Користувач вводить некоректні логін/пароль, тисне кнопку логіну і отримує повідомлення про неправильність введених даних	Співпадає з очікуваним
5	Директор переглядає інформацію про аптечний заклад	Директор тисне на кнопку деталей про аптечний заклад та отримує сторінку з детальним описом аптечного закладу, якщо поля заповнені.	Співпадає з очікуваним
6	Директор переглядає інформацію про працівників аптечного закладу	Директор тисне на кнопку «Фармацевти» у головному меню сайту та потрапляє на сторінку зі списком працівників аптечного закладу	Співпадає з очікуваним

Продовження таблиці 5.1

7	Директор переглядає інформацію про фармацевта	Директор натискає натискає на фармацевта та отримує сторінку з детальним описом фармацевта	Співпадає з очікуваним
8	Директор намагається створити фармацевта	Директор тисне на кнопку «Додати фармацевта» та переходить на сторінку з формою реєстрації фармацевта	Співпадає з очікуваним
9	Директор переглядає звіт погашених електронних рецептів аптечного закладу	Директор тисне на кнопку «Звіт» у головному меню системи та потрапляє на сторінку з формою для формування звіту	Співпадає з очікуваним
10	Користувач намагається змінити пароль	Користувач натискає на кнопку редагування профілю та отримує сторінку з формою для зміни паролю	Співпадає з очікуваним
11	Користувач намагається відновити пароль	Користувач натискає кнопку відновлення паролю і переходить на сторінку з формою відновлення паролю	Співпадає з очікуваним
12	Фармацевт переглядає електронний рецепт у системі	Фармацевт вводить в поле пошуку рецепту номер та переходить до сторінки з детальною інформацією про рецепт	Співпадає з очікуваним

Висновок до розділу

У розділі представлено керівництво користувача, яке включає опис дій, які можуть виконувати різні типи користувачів на сайті, а саме: авторизація, реєстрація, перегляд інформаційних сторінок сайту, редагування контенту, погашення електронних рецептів.

Розділ містить зображення екранних форм веб-застосування аптечного закладу і опис роботи системи на цих формах для правильного використання системи.

Представлені сценарії тестування веб-застосування на відповідність функціональним вимогам. Всі тести були успішно пройдені веб-застосуванням і збіглися з функціональними вимогами до системи.

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		51

ЗАГАЛЬНІ ВИСНОВКИ

В ході реалізації дипломного проекту була досліджена предметна область, визначені вимоги до веб-застосування. Вхідні дані та їх джерела визначені. Вихідні дані веб-застосування визначені, надана їх структура. Описано і відображено процеси і функції, які реалізовані в веб-застосуванні за допомогою діаграм діяльності та варіантів використання.

На підставі даних, отриманих в ході аналізу, була сформульована задача розробки веб-застосування. Розроблено і виділено основні цілі, які повинні бути реалізовані у веб-застосуванні аптечного закладу, а також визначені завдання і функціональні вимоги до системи.

Для розробки програмного забезпечення використовувалися наступні інструменти: мови C # і TypeScript, Entity Framework, AngularJS і ASP.NET MVC 4. Детальний опис інструментів розробки і список основних переваг обраних інструментів над їх аналогами.

У дипломному проекті розроблена гнучка програмна архітектура, яка підтримує розширення і розрахована на довгострокову роботу з підтримкою сучасних тенденцій в розробці програмного забезпечення.

Була розроблена модель бази даних, яка дозволяє зберігати і отримувати доступ до даних комплексу задач. СУБД MS SQL використовується для управління базою даних. Схема бази даних була розроблена з перспективою розширення функціональності системи.

Наведено інструкцію користувача по роботі веб-застосування аптечного закладу. Описані кроки необхідні для виконання всіх функцій веб-застосування, надаються копії екранних форм. Описано тести програмного продукту на відповідність функціональним вимогам, які система успішно пройшла.

ПЕРЕЛІК ПОСИЛАНЬ

1. MS SQL – інформація про СУБД MS SQL [Електронний ресурс] – Режим доступу до ресурса: <https://www.microsoft.com/ru-ru/sql-server/sql-server-2019>.
2. ASP.NET – сторінка бібліотеки ASP.NET [Електронний ресурс] – Режим доступу до ресурса: <http://www.asp.net/get-started>.
3. Entity Framework – інструкція побудови зав'язків за допомогою бібліотеки Entity Framework [Електронний ресурс] – Режим доступу до ресурса: <https://msdn.microsoft.com/en-us/data/jj713564.aspx>.
4. Тренд – інформація про визначення тренду [Електронний ресурс] – Режим доступу до ресурса: <https://chto-eto-takoe.ru/trend>.
5. Лінія тренду – інформація про визначення лінії тренду [Електронний ресурс] – Режим доступу до ресурса: <https://exceltable.com/grafiki/liniya-trenda-v-excel>.
6. С# – інформація про мову програмування С# [Електронний ресурс] – Режим доступу до ресурса: <https://docs.microsoft.com/ru-ru/dotnet/csharp/language-reference/operators/>.

Додаток А

Тексти програмного коду
WEB-ЗАСТОСУВАННЯ ДЛЯ ПІДСИСТЕМИ ПРОДАЖУ ТА ОБЛІКУ
ФАРМАЦЕВТИЧНИХ ПРЕПАРАТІВ КОРПОРАТИВНОЇ ERP
СИСТЕМИ

(Найменування програми (документа))

DVD-R

(Вид носія даних)

20 арк, 4 070 Кб

(Обсяг програми (документа) , арк.) Кб)

					ДП ІС-6223.00.000.ПЗ	Арк.
Змн.	Арк.	№ докум.	Підпис	Дата		54

Код веб-застосування

```
using System;
using System.Collections.Generic;
using System.Diagnostics;
using System.Linq;
using System.Linq.Expressions;
using System.Runtime.CompilerServices;
using System.Runtime.InteropServices;
using System.Threading.Tasks;

namespace Ph.Bl.Services
{
    public class EhealthWebService : CommonWebService
    {
        public static EhealthWebService Instance = new EhealthWebService();
        private static ILogger \u0002 = CommonService.Instance.Serilog;
        private Dictionary<long, EhealthToken> \u0003;
        private ServiceFactory \u0005;
        private RepositoryFactory \u0008;

        protected EhealthWebService()
        {
        }

        private Dictionary<long, EhealthToken> \u0002()
        {
            return this.\u0003;
        }

        private void \u0003(Dictionary<long, EhealthToken> _param1)
        {
            this.\u0003 = _param1;
        }

        public Task<JObject> QualifyMedicationRequestbyIdOneRegistryUsingES(string id)
        {
            EhealthWebService.\u0006\u2003\u2000 stateMachine;
            stateMachine.\u0005 = this;
            stateMachine.\u0008 = id;
            stateMachine.\u0003 = AsyncTaskMethodBuilder<JObject>.Create();
            stateMachine.\u0002 = -1;
            stateMachine.\u0003.Start<EhealthWebService.\u0006\u2003\u2000>(ref stateMachine);
            return stateMachine.\u0003.Task;
        }
    }
}
```

Змн.	Арк.	№ докум.	Підпис	Дата

```
public Task<JObject> GetLegalEntityV2DetailsAsync()
{
    EhealthWebService.\u0006\u2001\u2000 stateMachine;
    stateMachine.\u0005 = this;
    stateMachine.\u0003 = AsyncTaskMethodBuilder<JObject>.Create();
    stateMachine.\u0002 = -1;
    stateMachine.\u0003.Start<EhealthWebService.\u0006\u2001\u2000>(ref stateMachine);
    return stateMachine.\u0003.Task;
}

public Task<JObject> GetEmployeeDetailsAsync(string type)
{
    EhealthWebService.\u0005\u2001\u2000 stateMachine;
    stateMachine.\u0005 = this;
    stateMachine.\u0008 = type;
    stateMachine.\u0003 = AsyncTaskMethodBuilder<JObject>.Create();
    stateMachine.\u0002 = -1;
    stateMachine.\u0003.Start<EhealthWebService.\u0005\u2001\u2000>(ref stateMachine);
    return stateMachine.\u0003.Task;
}

public EhealthToken GetEhealthCurrentToken(long userId)
{
    string key = this.\u0008(userId);
    EhealthToken ehealthToken = this.ServiceFactory.RedisService.Get<EhealthToken>(key);
    if (ehealthToken == null)
    {
        ehealthToken = this.\u0005(userId);
        try
        {
            if (ehealthToken != null)
            {
                int result;

                int.TryParse(this.ServiceFactory.AppServiceConfiguration.Get(\u0008\u2004\u2001.\u0002(1930920
647)), out result);
                this.ServiceFactory.RedisService.Set<EhealthToken>(key, ehealthToken, result > 0 ?
result : 4);
            }
        }
        catch
        {
        }
    }
}
```

```
        return ehealthToken;
    }

    private string \u0008(long _param1)
    {
        string str =
this.ServiceFactory.AppServiceConfiguration.Get(\u0008\u2004\u2001.\u0002(1930908097));
        return string.Format(\u0008\u2004\u2001.\u0002(1930920619), (object)
\u0008\u2004\u2001.\u0002(1930920599), (object) _param1, (object) str);
    }

    public EhealthToken GetRefreshToken(long UserId)
    {
        return
this.ServiceFactory.RedisService.Get<EhealthToken>(\u0008\u2004\u2001.\u0002(1930920579) +
UserId.ToString());
    }

    public void AddEhealthToken(EhealthToken token, long UserId)
    {
        if (this.\u0002() == null)
            this.\u0003(new Dictionary<long, EhealthToken>());
        if (this.\u0002().ContainsKey(UserId))
            this.\u0002().Remove(UserId);
        this.\u0002().Add(UserId, token);
    }

    public void RemoveEhealthToken(long userId)
    {
        this.\u0006(userId);
        try
        {
            this.ServiceFactory.RedisService.Remove<string>(this.\u0008(userId));

this.ServiceFactory.RedisService.Remove<EhealthToken>(\u0008\u2004\u2001.\u0002(1930920579) +
(object) userId);
        }
        catch
        {
        }
    }

    public bool TryRemoveTokenByUserId(long userId)
    {
        bool flag = false;
```

Змн.	Арк.	№ докум.	Підпис	Дата

```
try
{
    this.\u0006(userId);
    try
    {
        this.ServiceFactory.RedisService.Remove<string>(this.\u0008(userId));
    }
    catch
    {
    }
    return true;
}
catch (Exception ex)
{
    return flag;
}

public ServiceFactory ServiceFactory
{
    get
    {
        if (this.\u0005 == null)
            this.\u0005 = ServiceFactory.Instanse;
        return this.\u0005;
    }
    set
    {
        this.\u0005 = value;
    }
}

public RepositoryFactory RepositoryFactory
{
    get
    {
        if (this.\u0008 == null)
            this.\u0008 = RepositoryFactory.Instance;
        return this.\u0008;
    }
    set
    {
        this.\u0008 = value;
    }
}
```

Змн.	Арк.	№ докум.	Підпис	Дата

```

public static string PrepareUrlParameters(params string[] parameters)
{
    parameters = ((IEnumerable<string>)
parameters).Where<string>(EhealthWebService.\u000E\u2000.\u0003 ??
(EhealthWebService.\u000E\u2000.\u0003 = new Func<string,
bool>(EhealthWebService.\u000E\u2000.\u0002.\u0002))).ToArray<string>());
    return !\u0005\u2002\u2001.\u0002<string>(parameters) ? string.Empty :
\u0008\u2004\u2001.\u0002(1930920815) + string.Join(\u0008\u2004\u2001.\u0002(1930920807),
parameters);
}

public async Task<string> ConfirmLegal(EqFile file)
{
    EhealthWebService ehealthWebService = this;
    object obj = (object) string.Empty;
    try
    {
        obj = await ehealthWebService.PostAsync<object>(\u0008\u2004\u2001.\u0002(1930925561),
CommonWebService.AccessDataMap[\u0008\u2004\u2001.\u0002(1930925561)].apiPath +
\u0008\u2004\u2001.\u0002(1930922668), (object) file, (object) null, (Dictionary<string,
string>) null, true);
        if (EhealthWebService.\u0002\u2009.\u0003 == null)
            EhealthWebService.\u0002\u2009.\u0003 = CallSite<Func<CallSite, object,
JObject>>.Create(Binder.Convert(CSharpBinderFlags.None, typeof (JObject), typeof
(EhealthWebService)));
        Func<CallSite, object, JObject> target = EhealthWebService.\u0002\u2009.\u0003.Target;
        CallSite<Func<CallSite, object, JObject>> callSite =
EhealthWebService.\u0002\u2009.\u0003;
        if (EhealthWebService.\u0002\u2009.\u0002 == null)
            EhealthWebService.\u0002\u2009.\u0002 = CallSite<Func<CallSite, System.Type, object,
object>>.Create(Binder.InvokeMember(CSharpBinderFlags.None,
\u0008\u2004\u2001.\u0002(1930909130), (IEnumerable<System.Type>) null, typeof
(EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[2]
{
            CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType |
CSharpArgumentInfoFlags.IsStaticType, (string) null),
            CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
        }));
        object obj1 = EhealthWebService.\u0002\u2009.\u0002.Target((CallSite)
EhealthWebService.\u0002\u2009.\u0002, typeof (JObject), obj);
        JObject jobject = target((CallSite) callSite, obj1);
        if (jobject[\u0008\u2004\u2001.\u0002(1930908471)] == null)
        {

```

```

        string str = jobject[\u0008\u2004\u2001.\u0002(1930904400)][(object)
\u0008\u2004\u2001.\u0002(1930922649)][(object)
\u0008\u2004\u2001.\u0002(1930922638)].ToString();

        string str1 = jobject[\u0008\u2004\u2001.\u0002(1930904400)][(object)
\u0008\u2004\u2001.\u0002(1930922649)][(object)
\u0008\u2004\u2001.\u0002(1930922878)].ToString();

ehealthWebService.ServiceFactory.AppServiceConfiguration.Set(\u0008\u2004\u2001.\u0002(1930908
097), str);

ehealthWebService.ServiceFactory.AppServiceConfiguration.Set(\u0008\u2004\u2001.\u0002(1930905
885), str1);

        await ehealthWebService.CreateFilial();
        ehealthWebService.\u0008\u2000(str);
        str = (string) null;
    }
    if (EhealthWebService.\u0002\u2009.\u0005 == null)
        EhealthWebService.\u0002\u2009.\u0005 = CallSite<Func<CallSite, object,
string>>.Create(Binder.Convert(CSharpBinderFlags.None, typeof (string), typeof
(EhealthWebService)));
        return EhealthWebService.\u0002\u2009.\u0005.Target((CallSite)
EhealthWebService.\u0002\u2009.\u0005, obj);
    }
    catch (Exception ex)
    {

ehealthWebService.ServiceFactory.CommonService.Log(\u0008\u2004\u2001.\u0002(1930922858) +
ex.Message, \u0008\u2004\u2001.\u0002(1930922828), new bool?(false),
ehealthWebService.ServiceFactory.AppServiceConfiguration.GetIServerVariables());
        return ex.Message;
    }
}

public Task CreateFilial()
{
    EhealthWebService.\u0005\u2000\u2000 stateMachine;
    stateMachine.\u0005 = this;
    stateMachine.\u0003 = AsyncTaskMethodBuilder.Create();
    stateMachine.\u0002 = -1;
    stateMachine.\u0003.Start<EhealthWebService.\u0005\u2000\u2000>(ref stateMachine);
    return stateMachine.\u0003.Task;
}

public void AddConfig(System.Configuration.Configuration config, string key, string value)
{

```



```

        if (((IEnumerable<string>) config.AppSettings.Settings.AllKeys).Contains<string>(key))
            config.AppSettings.Settings.Remove(key);
        config.AppSettings.Settings.Add(key, value);
    }

    public void SaveAttempt(StrHolder str)
    {
        Task.Run(new Func<Task>(new EhealthWebService.\u0005\u2001()
        {
            \u0002 = this,
            \u0003 = str
        }).\u0002)).Wait();
    }

    public object GetDictionaries()
    {
        EhealthWebService.\u0008\u2001 obj1 = new EhealthWebService.\u0008\u2001();
        obj1.\u0003 = this;
        obj1.\u0002 = (object) null;
        Task.Run(new Func<Task>(obj1.\u0002)).Wait();
        object obj2 = obj1.\u0002;
        if (EhealthWebService.\u0003\u2009.\u0002 == null)
            EhealthWebService.\u0003\u2009.\u0002 = CallSite<Func<CallSite, object,
object>>.Create(Binder.GetMember(CSharpBinderFlags.None,
\u0008\u2004\u2001.\u0002(1930908459), typeof (EhealthWebService),
(IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[1]
        {
            CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
        }));
        return EhealthWebService.\u0003\u2009.\u0002.Target((CallSite)
EhealthWebService.\u0003\u2009.\u0002, obj2);
    }

    public eRegion[] GetRegions()
    {
        EhealthWebService.\u0006\u2001 obj1 = new EhealthWebService.\u0006\u2001();
        obj1.\u0003 = this;
        obj1.\u0002 = (object) null;
        Task.Run(new Func<Task>(obj1.\u0002)).Wait();
        object obj2 = obj1.\u0002;
        if (EhealthWebService.\u0005\u2009.\u0003 == null)
            EhealthWebService.\u0005\u2009.\u0003 = CallSite<Func<CallSite, EhealthWebService,
object, object>>.Create(Binder.InvokeMember(CSharpBinderFlags.InvokeSimpleName,
\u0008\u2004\u2001.\u0002(1930920799), (IEnumerable<System.Type>) new System.Type[1]
        {

```

```

        typeof (eRegion[]))
    }, typeof (EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new
CSharpArgumentInfo[2]
    {
        CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType, (string)
null),
        CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
    }));
    Func<CallSite, EhealthWebService, object, object> target =
EhealthWebService.\u0005\u2009.\u0003.Target;
    CallSite<Func<CallSite, EhealthWebService, object, object>> callSite =
EhealthWebService.\u0005\u2009.\u0003;
    if (EhealthWebService.\u0005\u2009.\u0002 == null)
        EhealthWebService.\u0005\u2009.\u0002 = CallSite<Func<CallSite, object,
object>>.Create(Binder.GetMember(CSharpBinderFlags.None,
\u0008\u2004\u2001.\u0002(1930908459), typeof (EhealthWebService),
(IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[1]
    {
        CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
    }));
    object obj3 = EhealthWebService.\u0005\u2009.\u0002.Target((CallSite)
EhealthWebService.\u0005\u2009.\u0002, obj2);
    return ((IEnumerable<eRegion>) (target((CallSite) callSite, this, obj3) as
eRegion[])).OrderBy<eRegion, string>(EhealthWebService.\u000E\u2000.\u0005 ??
(EhealthWebService.\u000E\u2000.\u0005 = new Func<eRegion,
string>(EhealthWebService.\u000E\u2000.\u0002.\u0002))).ToArray<eRegion>();
}

public eDistricts[] GetDistricts(string regionName)
{
    EhealthWebService.\u000E\u2001 obj1 = new EhealthWebService.\u000E\u2001();
    obj1.\u0003 = this;
    obj1.\u0005 = regionName;
    obj1.\u0002 = (object) null;
    Task.Run(new Func<Task>(obj1.\u0002)).Wait();
    object obj2 = obj1.\u0002;
    if (EhealthWebService.\u0008\u2009.\u0005 == null)
        EhealthWebService.\u0008\u2009.\u0005 = CallSite<Func<CallSite, object,
eDistricts[]>>.Create(Binder.Convert(CSharpBinderFlags.None, typeof (eDistricts[]), typeof
(EhealthWebService)));
    Func<CallSite, object, eDistricts[]> target1 =
EhealthWebService.\u0008\u2009.\u0005.Target;
    CallSite<Func<CallSite, object, eDistricts[]>> callSite1 =
EhealthWebService.\u0008\u2009.\u0005;
    if (EhealthWebService.\u0008\u2009.\u0003 == null)

```

```

        EhealthWebService.\u0008\u2009.\u0003 = CallSite<Func<CallSite, EhealthWebService,
object, object>>.Create(Binder.InvokeMember(CSharpBinderFlags.InvokeSimpleName,
\u0008\u2004\u2001.\u0002(1930920784), (IEnumerable<System.Type>) new System.Type[1]
    {
        typeof (eDistricts[])
    }, typeof (EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new
CSharpArgumentInfo[2]
    {
        CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType, (string)
null),
        CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
    });
    Func<CallSite, EhealthWebService, object, object> target2 =
EhealthWebService.\u0008\u2009.\u0003.Target;
    CallSite<Func<CallSite, EhealthWebService, object, object>> callSite2 =
EhealthWebService.\u0008\u2009.\u0003;
    if (EhealthWebService.\u0008\u2009.\u0002 == null)
        EhealthWebService.\u0008\u2009.\u0002 = CallSite<Func<CallSite, object,
object>>.Create(Binder.GetMember(CSharpBinderFlags.None,
\u0008\u2004\u2001.\u0002(1930908459), typeof (EhealthWebService),
(IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[1]
    {
        CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
    });
    object obj3 = EhealthWebService.\u0008\u2009.\u0002.Target((CallSite)
EhealthWebService.\u0008\u2009.\u0002, obj2);
    object obj4 = target2((CallSite) callSite2, this, obj3);
    return target1((CallSite) callSite1, obj4);
}

public eSettlement[] GetSettlements(
    string regionName,
    string districtName,
    string settlementName)
{
    EhealthWebService.\u000F\u2001 obj1 = new EhealthWebService.\u000F\u2001();
    obj1.\u0003 = this;
    obj1.\u0005 = regionName;
    obj1.\u0008 = settlementName;
    obj1.\u0006 = districtName;
    obj1.\u0002 = (object) null;
    Task.Run(new Func<Task>(obj1.\u0002)).Wait();
    object obj2 = obj1.\u0002;
    if (EhealthWebService.\u0006\u2009.\u0003 == null)

```

```

EhealthWebService.\u0006\u2009.\u0003 = CallSite<Func<CallSite, EhealthWebService,
object, object>>.Create(Binder.InvokeMember(CSharpBinderFlags.InvokeSimpleName,
\u0008\u2004\u2001.\u0002(1930920773), (IEnumerable<System.Type>) new System.Type[1]
{
    typeof (eSettlement[])
}, typeof (EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new
CSharpArgumentInfo[2]
{
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType, (string)
null),
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
}));
Func<CallSite, EhealthWebService, object, object> target =
EhealthWebService.\u0006\u2009.\u0003.Target;
CallSite<Func<CallSite, EhealthWebService, object, object>> callSite =
EhealthWebService.\u0006\u2009.\u0003;
if (EhealthWebService.\u0006\u2009.\u0002 == null)
    EhealthWebService.\u0006\u2009.\u0002 = CallSite<Func<CallSite, object,
object>>.Create(Binder.GetMember(CSharpBinderFlags.None,
\u0008\u2004\u2001.\u0002(1930908459), typeof (EhealthWebService),
(IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[1]
{
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
}));
object obj3 = EhealthWebService.\u0006\u2009.\u0002.Target((CallSite)
EhealthWebService.\u0006\u2009.\u0002, obj2);
return ((IEnumerable<eSettlement>) (target((CallSite) callSite, this, obj3) as
eSettlement[])).Where<eSettlement>(new Func<eSettlement,
bool>(obj1.\u0002)).OrderBy<eSettlement, string>(EhealthWebService.\u000E\u2000.\u0008 ??
(EhealthWebService.\u000E\u2000.\u0008 = new Func<eSettlement,
string>(EhealthWebService.\u000E\u2000.\u0002.\u0002))).ThenBy<eSettlement,
string>(EhealthWebService.\u000E\u2000.\u0006 ?? (EhealthWebService.\u000E\u2000.\u0006 = new
Func<eSettlement,
string>(EhealthWebService.\u000E\u2000.\u0002.\u0003))).ToArray<eSettlement>());
}

public CompanyVm FindCompany(string erdpou)
{
    EhealthWebService.\u0002\u2002 obj = new EhealthWebService.\u0002\u2002();
    obj.\u0002 = this;
    obj.\u0003 = erdpou;
    CommonWebService.freeFire = true;
    CompanyVm result = Task.Run<CompanyVm>(new Func<Task<CompanyVm>>(obj.\u0002)).Result;
    CommonWebService.freeFire = false;
    return result;
}

```

					ДП ІС-6223.00.000.ПЗ	Арк.
						64
Змн.	Арк.	№ докум.	Підпис	Дата		

}

public object ExchangeGrantToken(

string code,

string client_id,

string client_secret,

string redirect_uri,

string scope)

{

return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0003\u2002()

{

\u0002 = this,

\u0003 = code,

\u0005 = client_id,

\u0008 = client_secret,

\u0006 = redirect_uri,

\u000E = scope

}).\u0002)).Result;

}

public object ExchangeGrantTokenEhealthTemp(

string code,

string client_id,

string client_secret,

string redirect_uri,

string scope,

string userId)

{

EhealthWebService.\u0005\u2002 obj = new EhealthWebService.\u0005\u2002()

{

\u0002 = this,

\u0003 = new \u000F\u2009<string, string, string, string, string,

string>(\u0008\u2004\u2001.\u0002(1930925569), code, client_id, client_secret, redirect_uri,
scope),

\u0005 = new Dictionary<string, string>()

};

obj.\u0005.Add(\u0008\u2004\u2001.\u0002(1930909174), client_id);

obj.\u0005.Add(\u0008\u2004\u2001.\u0002(1930909159), userId);

object result = Task.Run<object>(new Func<Task<object>>(obj.\u0002)).Result;

if (EhealthWebService.\u000E\u2009.\u0002 == null)

EhealthWebService.\u000E\u2009.\u0002 = CallSite<Action<CallSite, EhealthWebService,
object>>.Create(Binder.InvokeMember(CSharpBinderFlags.InvokeSimpleName |
CSharpBinderFlags.ResultDiscarded, \u0008\u2004\u2001.\u0002(1930920766),
(IEnumerable<System.Type>) null, typeof (EhealthWebService), (IEnumerable<CSharpArgumentInfo>)
new CSharpArgumentInfo[2]

```

        {
            CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType, (string)
null),
            CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
        });
        EhealthWebService.\u000E\u2009.\u0002.Target((CallSite)
EhealthWebService.\u000E\u2009.\u0002, this, result);
        return result;
    }

    private void \u000E(object _param1)
    {
        int minutes = 20156;
        if (EhealthWebService.\u000F\u2009.\u0002 == null)
            EhealthWebService.\u000F\u2009.\u0002 = CallSite<Func<CallSite, object, string,
object>>.Create(Binder.GetIndex(CSharpBinderFlags.None, typeof (EhealthWebService),
(IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[2]
        {
            CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null),
            CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType |
CSharpArgumentInfoFlags.Constant, (string) null)
        }));
        object obj1 = EhealthWebService.\u000F\u2009.\u0002.Target((CallSite)
EhealthWebService.\u000F\u2009.\u0002, _param1, \u0008\u2004\u2001.\u0002(1930908459));
        object obj2;
        if (obj1 == null)
        {
            obj2 = (object) null;
        }
        else
        {
            if (EhealthWebService.\u000F\u2009.\u0003 == null)
                EhealthWebService.\u000F\u2009.\u0003 = CallSite<Func<CallSite, object, string,
object>>.Create(Binder.GetIndex(CSharpBinderFlags.None, typeof (EhealthWebService),
(IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[2]
                {
                    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null),
                    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType |
CSharpArgumentInfoFlags.Constant, (string) null)
                }));
            object obj3 = EhealthWebService.\u000F\u2009.\u0003.Target((CallSite)
EhealthWebService.\u000F\u2009.\u0003, obj1, \u0008\u2004\u2001.\u0002(1930906056));
            if (obj3 == null)
            {
                obj2 = (object) null;
            }
        }
    }

```

```

    }
    else
    {
        if (EhealthWebService.\u000F\u2009.\u0005 == null)
            EhealthWebService.\u000F\u2009.\u0005 = CallSite<Func<CallSite, object, string,
object>>.Create(Binder.GetIndex(CSharpBinderFlags.None, typeof (EhealthWebService),
(IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[2]
        {
            CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null),
            CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType |
CSharpArgumentInfoFlags.Constant, (string) null)
        });
        obj2 = EhealthWebService.\u000F\u2009.\u0005.Target((CallSite)
EhealthWebService.\u000F\u2009.\u0005, obj3, \u0008\u2004\u2001.\u0002(1930906076));
    }
}
object obj4 = obj2;
if (EhealthWebService.\u000F\u2009.\u0006 == null)
    EhealthWebService.\u000F\u2009.\u0006 = CallSite<Func<CallSite, object,
bool>>.Create(Binder.UnaryOperation(CSharpBinderFlags.None, ExpressionType.IsTrue, typeof
(EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[1]
    {
        CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
    }));
Func<CallSite, object, bool> target1 = EhealthWebService.\u000F\u2009.\u0006.Target;
CallSite<Func<CallSite, object, bool>> callSite1 =
EhealthWebService.\u000F\u2009.\u0006;
if (EhealthWebService.\u000F\u2009.\u0008 == null)
    EhealthWebService.\u000F\u2009.\u0008 = CallSite<Func<CallSite, object, object,
object>>.Create(Binder.BinaryOperation(CSharpBinderFlags.None, ExpressionType.NotEqual, typeof
(EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[2]
    {
        CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null),
        CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.Constant, (string) null)
    }));
object obj5 = EhealthWebService.\u000F\u2009.\u0008.Target((CallSite)
EhealthWebService.\u000F\u2009.\u0008, obj4, (object) null);
if (!target1((CallSite) callSite1, obj5))
    return;
DateTime dateTime = DateTime.UtcNow.AddMinutes((double) minutes);
EhealthToken ehealthToken1 = new EhealthToken();
EhealthToken ehealthToken2 = ehealthToken1;
if (EhealthWebService.\u000F\u2009.\u000F == null)

```

```

        EhealthWebService.\u000F\u2009.\u000F = CallSite<Func<CallSite, object,
string>>.Create(Binder.Convert(CSharpBinderFlags.None, typeof (string), typeof
(EhealthWebService)));

        Func<CallSite, object, string> target2 = EhealthWebService.\u000F\u2009.\u000F.Target;
        CallSite<Func<CallSite, object, string>> callSite2 =
EhealthWebService.\u000F\u2009.\u000F;
        if (EhealthWebService.\u000F\u2009.\u000E == null)
            EhealthWebService.\u000F\u2009.\u000E = CallSite<Func<CallSite, object,
object>>.Create(Binder.InvokeMember(CSharpBinderFlags.None,
\u0008\u2004\u2001.\u0002(1930903853), (IEnumerable<System.Type>) null, typeof
(EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[1]
{
            CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
        }));
        object obj6 = EhealthWebService.\u000F\u2009.\u000E.Target((CallSite)
EhealthWebService.\u000F\u2009.\u000E, obj4);
        string str = target2((CallSite) callSite2, obj6);
        ehealthToken2.Token = str;
        ehealthToken1.Finish = dateTime;
        this.ServiceFactory.RedisService.Set<EhealthToken>(\u0008\u2004\u2001.\u0002(1930920579)
+ this.ServiceFactory.AppService.currentUser.Id.ToString(), ehealthToken1, minutes);
    }

    public Task<object> GetEhealthAuthLinkAsync(string role, long phUserId)
    {
        EhealthWebService.\u0003\u2001\u2000 stateMachine;
        stateMachine.\u0005 = this;
        stateMachine.\u0006 = role;
        stateMachine.\u0008 = phUserId;
        stateMachine.\u0003 = AsyncTaskMethodBuilder<object>.Create();
        stateMachine.\u0002 = -1;
        stateMachine.\u0003.Start<EhealthWebService.\u0003\u2001\u2000>(ref stateMachine);
        return stateMachine.\u0003.Task;
    }

    public Task<JObject> GetUserDetailsAsync()
    {
        EhealthWebService.\u0005\u2002\u2000 stateMachine;
        stateMachine.\u0005 = this;
        stateMachine.\u0003 = AsyncTaskMethodBuilder<JObject>.Create();
        stateMachine.\u0002 = -1;
        stateMachine.\u0003.Start<EhealthWebService.\u0005\u2002\u2000>(ref stateMachine);
        return stateMachine.\u0003.Task;
    }

```



```

public object GetEhealthOwners(string token, string legalId, int page = 1)
{
    object ehealthEmployees = this.GetEhealthEmployees(token, legalId,
    \u0008\u2004\u2001.\u0002(1930905959), page);
    if (EhealthWebService.\u0002\u200A.\u000F == null)
        EhealthWebService.\u0002\u200A.\u000F = CallSite<Func<CallSite, object,
bool>>.Create(Binder.UnaryOperation(CSharpBinderFlags.None, ExpressionType.IsTrue, typeof
(EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[1]
{
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
}));
    Func<CallSite, object, bool> target1 = EhealthWebService.\u0002\u200A.\u000F.Target;
    CallSite<Func<CallSite, object, bool>> callSite1 =
EhealthWebService.\u0002\u200A.\u000F;
    if (EhealthWebService.\u0002\u200A.\u0003 == null)
        EhealthWebService.\u0002\u200A.\u0003 = CallSite<Func<CallSite, object, object,
object>>.Create(Binder.BinaryOperation(CSharpBinderFlags.None, ExpressionType.NotEqual, typeof
(EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[2]
{
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null),
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.Constant, (string) null)
}));
    Func<CallSite, object, object, object> target2 =
EhealthWebService.\u0002\u200A.\u0003.Target;
    CallSite<Func<CallSite, object, object, object>> callSite2 =
EhealthWebService.\u0002\u200A.\u0003;
    if (EhealthWebService.\u0002\u200A.\u0002 == null)
        EhealthWebService.\u0002\u200A.\u0002 = CallSite<Func<CallSite, object, string,
object>>.Create(Binder.GetIndex(CSharpBinderFlags.None, typeof (EhealthWebService),
(IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[2]
{
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null),
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType |
CSharpArgumentInfoFlags.Constant, (string) null)
}));
    object obj1 = EhealthWebService.\u0002\u200A.\u0002.Target((CallSite)
EhealthWebService.\u0002\u200A.\u0002, ehealthEmployees,
    \u0008\u2004\u2001.\u0002(1930908459));
    object obj2 = target2((CallSite) callSite2, obj1, (object) null);
    if (EhealthWebService.\u0002\u200A.\u000E == null)
        EhealthWebService.\u0002\u200A.\u000E = CallSite<Func<CallSite, object,
bool>>.Create(Binder.UnaryOperation(CSharpBinderFlags.None, ExpressionType.IsFalse, typeof
(EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[1]
{
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
}));

```

Змн.	Арк.	№ докум.	Підпис	Дата

```

    }));
    object obj3;
    if (!EhealthWebService.\u0002\u200A.\u000E.Target((CallSite)
EhealthWebService.\u0002\u200A.\u000E, obj2))
    {
        if (EhealthWebService.\u0002\u200A.\u0006 == null)
            EhealthWebService.\u0002\u200A.\u0006 = CallSite<Func<CallSite, object, bool,
object>>.Create(Binder.BinaryOperation(CSharpBinderFlags.BinaryOperationLogical,
ExpressionType.And, typeof (EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new
CSharpArgumentInfo[2]
            {
                CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null),
                CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType, (string)
null)
            }
        ));
        Func<CallSite, object, bool, object> target3 =
EhealthWebService.\u0002\u200A.\u0006.Target;
        CallSite<Func<CallSite, object, bool, object>> callSite3 =
EhealthWebService.\u0002\u200A.\u0006;
        object obj4 = obj2;
        if (EhealthWebService.\u0002\u200A.\u0008 == null)
            EhealthWebService.\u0002\u200A.\u0008 = CallSite<Func<CallSite, object,
JArray>>.Create(Binder.Convert(CSharpBinderFlags.ConvertExplicit, typeof (JArray), typeof
(EhealthWebService)));
        Func<CallSite, object, JArray> target4 = EhealthWebService.\u0002\u200A.\u0008.Target;
        CallSite<Func<CallSite, object, JArray>> callSite4 =
EhealthWebService.\u0002\u200A.\u0008;
        if (EhealthWebService.\u0002\u200A.\u0005 == null)
            EhealthWebService.\u0002\u200A.\u0005 = CallSite<Func<CallSite, object, string,
object>>.Create(Binder.GetIndex(CSharpBinderFlags.None, typeof (EhealthWebService),
(IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[2]
            {
                CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null),
                CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType |
CSharpArgumentInfoFlags.Constant, (string) null)
            }
        ));
        object obj5 = EhealthWebService.\u0002\u200A.\u0005.Target((CallSite)
EhealthWebService.\u0002\u200A.\u0005, ehealthEmployees,
\u0008\u2004\u2001.\u0002(1930908459));
        int num = target4((CallSite) callSite4, obj5).Count > 0 ? 1 : 0;
        obj3 = target3((CallSite) callSite3, obj4, num != 0);
    }
    else
        obj3 = obj2;

```

```
return target1((CallSite) callSite1, obj3) ? ehealthEmployees :
this.GetEhealthEmployees(token, legalId, \u0008\u2004\u2001.\u0002(1930920755), page);
}

public object GetEhealthDoctors(string token, string legalId, int page = 1)
{
    return this.GetEhealthEmployees(token, legalId, \u0008\u2004\u2001.\u0002(1930906432),
page);
}

public object GetEhealthPharmacists(
    string token,
    string division_id,
    string legalId,
    int page = 1)
{
    return this.GetEmployeesList(token, division_id, legalId,
\u0008\u2004\u2001.\u0002(1930905981), page);
}

public object GetEhealthEmployees(string token, string legalId, string type, int page = 1)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0008\u2002()
{
    \u0002 = this,
    \u0003 = legalId,
    \u0005 = type,
    \u0008 = page,
    \u0006 = token
}).\u0002)).Result;
}

public object GetEmployeesList(
    string token,
    string division_id,
    string legalId,
    string type,
    int page = 1)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0006\u2002()
{
    \u0002 = this,
    \u0003 = division_id,
    \u0005 = legalId,
    \u0008 = type,
```

```
\u0006 = page,
\u000E = token
}).\u0002)).Result;
}

public object GetEhealthEmployers(string token, string legalId)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u000E\u2002()
    {
        \u0002 = this,
        \u0003 = legalId,
        \u0005 = token
    }).\u0002)).Result;
}

public object GetEhealthEmployeeDetail(string token, string id)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u000F\u2002()
    {
        \u0002 = this,
        \u0003 = id,
        \u0005 = token
    }).\u0002)).Result;
}

public object FindVerificationByPhone(string phone, string token)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0002\u2003()
    {
        \u0002 = this,
        \u0003 = phone,
        \u0005 = token
    }).\u0002)).Result;
}

public object InitializeVerificationByPhone(string phone, string token)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0003\u2003()
    {
        \u0002 = this,
        \u0003 = phone,
        \u0005 = token
    }).\u0002)).Result;
}
}
```

Змн.	Арк.	№ докум.	Підпис	Дата

```

public object CompleteVerificationByPhone(string phone, int? smsCode, string token)
{
    EhealthWebService.\u0005\u2003 obj = new EhealthWebService.\u0005\u2003()
    {
        \u0002 = this,
        \u0003 = phone,
        \u0008 = token,
        \u0005 = (object) null
    };
    obj.\u0005 = smsCode.HasValue ? (object) new \u000E\u200A<int?>(smsCode) : (object) new
    \u0006\u2001\u2000();
    return Task.Run<object>(new Func<Task<object>>(obj.\u0002)).Result;
}

public object CreateOrUpdateEhealthEntity(
    string body,
    string entityId,
    string token,
    EhealthEntityType type)
{
    EhealthWebService.\u0008\u2003 obj = new EhealthWebService.\u0008\u2003();
    obj.\u0002 = type;
    obj.\u0003 = this;
    obj.\u000E = token;
    obj.\u000F = entityId;
    obj.\u0005 = string.Empty;
    switch (obj.\u0002)
    {
        case EhealthEntityType.employee:
            obj.\u0005 = \u0008\u2004\u2001.\u0002(1930920727);
            break;
        case EhealthEntityType.division:
            obj.\u0005 = \u0008\u2004\u2001.\u0002(1930920743);
            break;
        case EhealthEntityType.patient:
            obj.\u0005 = \u0008\u2004\u2001.\u0002(1930920959);
            break;
    }
    obj.\u0008 = JsonConvert.DeserializeObject<object>(body);
    obj.\u0006 = this.\u0006\u2000();
    return obj.\u000F == null || obj.\u0002 == EhealthEntityType.employee ?
    Task.Run<object>(new Func<Task<object>>(obj.\u0002)).Result : Task.Run<object>(new
    Func<Task<object>>(obj.\u0003)).Result;
}

```

					ДП ІС-6223.00.000.ПЗ	Арк.
						73
Змн.	Арк.	№ докум.	Підпис	Дата		

```

public object ApproveEmployeeRequest(string ehealthId, string token)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0006\u2003()
    {
        \u0002 = this,
        \u0003 = ehealthId,
        \u0005 = token
    }).\u0002)).Result;
}

public object DismissEmployee(string ehealthId, string token)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u000E\u2003()
    {
        \u0002 = this,
        \u0003 = ehealthId,
        \u0005 = token
    }).\u0002)).Result;
}

public object GetDeclarationRequestsById(string token, string declarationId)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u000F\u2003()
    {
        \u0002 = this,
        \u0003 = declarationId,
        \u0005 = token
    }).\u0002)).Result;
}

public object SignDeclarationRequests(string token, string data, string declarationId)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0002\u2004()
    {
        \u0002 = this,
        \u0003 = declarationId,
        \u0005 = data,
        \u0008 = token
    }).\u0002)).Result;
}

public object ResendOtp(string token, string declarationId)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0003\u2004()
    {

```

```
\u0002 = this,
\u0003 = declarationId,
\u0005 = token
}).\u0002)).Result;
}

public object ApproveDeclaration(string token, string declarationId, int? smsCode)
{
    EhealthWebService.\u0005\u2004 obj = new EhealthWebService.\u0005\u2004()
    {
        \u0002 = this,
        \u0003 = declarationId,
        \u0008 = token,
        \u0005 = (object) null
    };
    obj.\u0005 = smsCode.HasValue ? (object) new \u0002<int?>(smsCode) : (object) new
    \u0006\u2001\u2000();
    return Task.Run<object>(new Func<Task<object>>(obj.\u0002)).Result;
}

public object GetDivisions(string token, string legalId, string id, int page = 1, int
PageSize = 500)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0008\u2004())
    {
        \u0002 = this,
        \u0003 = legalId,
        \u0005 = PageSize,
        \u0008 = page,
        \u0006 = id != null ? \u0008\u2004\u2001.\u0002(1930920928) + id : string.Empty,
        \u000E = this.\u0006\u2000()
    }).\u0002)).Result;
}

public object GetDivision(string token, string id)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0006\u2004())
    {
        \u0002 = this,
        \u0003 = id,
        \u0005 = this.\u0006\u2000()
    }).\u0002)).Result;
}

public object GetEmployeeRequest(string token, string id)
```

Змн.	Арк.	№ докум.	Підпис	Дата

```

{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u000E\u2004()
    {
        \u0002 = this,
        \u0003 = id,
        \u0005 = token
    }).\u0002)).Result;
}

public object GetEntity(string token, string id)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u000F\u2004()
    {
        \u0002 = this,
        \u0003 = id,
        \u0005 = token
    }).\u0002)).Result;
}

public object GetDeclaration(string token, string id)
{
    return Task.Run<object>(new Func<Task<object>>(new EhealthWebService.\u0002\u2005()
    {
        \u0002 = this,
        \u0003 = id,
        \u0005 = token
    }).\u0002)).Result;
}

public object GetDeclarationsList(string token, int page, int pageSize, string status)
{
    EhealthWebService.\u0003\u2005 obj = new EhealthWebService.\u0003\u2005();
    obj.\u0002 = this;
    obj.\u0005 = token;
    string str = EhealthWebService.PrepareUrlParameters(string.IsNullOrEmpty(status) ?
string.Empty : \u0008\u2004\u2001.\u0002(1930920184) + status,
string.Format(\u0008\u2004\u2001.\u0002(1930920917), (object) page, (object) pageSize));
    obj.\u0003 =
CommonWebService.AccessDataMap[\u0008\u2004\u2001.\u0002(1930925766)].apiPath +
\u0008\u2004\u2001.\u0002(1930920888) + str;
    return Task.Run<object>(new Func<Task<object>>(obj.\u0002)).Result;
}

public Dictionary<string, string> GetEhealthHead(string token)
{

```

Змн.	Арк.	№ докум.	Підпис	Дата


```

return new Dictionary<string, string>()
{
    {
        \u0008\u2004\u2001.\u0002(1930920865),
        \u0008\u2004\u2001.\u0002(1930920845) + token
    }
};
}

public T GetAs<T>(object value)
{
    if (EhealthWebService.\u0005\u200A<T>.\u0002 == null)
        EhealthWebService.\u0005\u200A<T>.\u0002 = CallSite<Func<CallSite, System.Type,
object, object>>.Create(Binder.InvokeMember(CSharpBinderFlags.None,
\u0008\u2004\u2001.\u0002(1930909118), (IEnumerable<System.Type>) null, typeof
(EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new CSharpArgumentInfo[2]
{
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType |
CSharpArgumentInfoFlags.IsStaticType, (string) null),
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
}));
    object obj1 = EhealthWebService.\u0005\u200A<T>.\u0002.Target((CallSite)
EhealthWebService.\u0005\u200A<T>.\u0002, typeof (JsonConvert), value);
    if (EhealthWebService.\u0005\u200A<T>.\u0005 == null)
        EhealthWebService.\u0005\u200A<T>.\u0005 = CallSite<Func<CallSite, object,
T>>.Create(Binder.Convert(CSharpBinderFlags.None, typeof (T), typeof (EhealthWebService)));
    Func<CallSite, object, T> target = EhealthWebService.\u0005\u200A<T>.\u0005.Target;
    CallSite<Func<CallSite, object, T>> callSite = EhealthWebService.\u0005\u200A<T>.\u0005;
    if (EhealthWebService.\u0005\u200A<T>.\u0003 == null)
        EhealthWebService.\u0005\u200A<T>.\u0003 = CallSite<Func<CallSite, System.Type,
object, object>>.Create(Binder.InvokeMember(CSharpBinderFlags.None,
\u0008\u2004\u2001.\u0002(1930906476), (IEnumerable<System.Type>) new System.Type[1]
{
    typeof (T)
}, typeof (EhealthWebService), (IEnumerable<CSharpArgumentInfo>) new
CSharpArgumentInfo[2]
{
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.UseCompileTimeType |
CSharpArgumentInfoFlags.IsStaticType, (string) null),
    CSharpArgumentInfo.Create(CSharpArgumentInfoFlags.None, (string) null)
}));
    object obj2 = EhealthWebService.\u0005\u200A<T>.\u0003.Target((CallSite)
EhealthWebService.\u0005\u200A<T>.\u0003, typeof (JsonConvert), obj1);
    return target((CallSite) callSite, obj2);
}

```

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
“КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО”

Кафедра автоматизованих систем обробки інформації і управління

УЗГОДЖЕНО

Керівник проєкту

_____ Олена ХАЛУС
(підпис) (вл. ім'я, прізвище)

“13” квітня 2020 р.

ЗАТВЕРДЖУЮ

В.о. завідувача кафедри

_____ Олександр ПАВЛОВ
(підпис) (вл. ім'я, прізвище)

“14” квітня 2020 р.

Підсистема продажу та обліку фармацевтичних препаратів
корпоративної ERP системи

ТЕХНІЧНЕ ЗАВДАННЯ

Шифр *ДП 6223.01.000 ТЗ*

на 11 сторінках

Київ – 2020 року

3MIST

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ.....	3
1.1 Повне найменування системи та її умовне позначення.....	3
1.2 Найменування організації-замовника та організаційучасників робіт.....	3
1.3 Перелік документів, на підставі яких створюється система	3
1.4. Планові терміни початку і закінчення роботи зі створення системи.....	4
2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ	5
2.1. Призначення системи.....	5
2.2. Цілі створення системи.....	5
3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ.....	6
4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	7
4.1. Вимоги до функціональних характеристик.....	7
4.2. Вимоги до надійності.....	8
4.3. Умови експлуатації	8
4.4. Вимоги до складу і параметрів технічних засобів.....	8
5 СТАДІЇ І ЕТАПИ РОЗРОБКИ	10
6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ.....	11
6.1. Види випробувань	11

					ДП 6223.01.000 ТЗ						
Зм.	Арк.	Прізвище	Підпис	Дата	Підсистема продажу та обліку фармацевтичних препаратів корпоративної ERP системи	Лім.	Лист	Листів			
Розроб.		Гаєвська А.О.									
Перевірів.		Халус О.А.					2	11			
						КПІ ім. Ігоря Сікорського Каф. АСОІУ Гр. ІС-361					
Н. кон.		Тєлешева Т.О.									
Затв.		Павлов О.А.									

1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

1.1 Повне найменування системи та її умовне позначення

Повне найменування системи: підсистема продажу та обліку фармацевтичних препаратів корпоративної ERP системи. Умовне позначення програмного продукту «pharmasu».

1.2 Найменування організації-замовника та організаційучасників робіт

Організацією-замовником виступає вищий навчальний заклад Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського», факультет інформатики та обчислювальної техніки, кафедра автоматизованих систем обробки інформації та управління (АСОІУ). Представником від замовника є старший викладач – Халус Олена Андріївна.

Виконавцем-розробником виступає студентка 4 курсу, факультету інформатики та обчислювальної техніки, групи ІС-361, Гаєвська Аліна Олегівна.

1.3 Перелік документів, на підставі яких створюється система

При розробці системи та створенні відповідної документації для програмного продукту, Виконавець обов'язково повинен дотримуватись вимог, які наведені у нормативних документах нижче:

- ДСТУ 19.201-78. Технічне завдання. Вимоги до змісту і оформлення;
- ДСТУ 34.201-89. Інформаційні технології. Комплекс стандартів на автоматизовані системи. Види, комплексність і позначення документів при створенні автоматизованих систем;

					ДП ІС-6223.01.000.ТЗ	Арк.
						3
Змн.	Арк.	№ докум.	Підпис	Дата		

– ДСТУ 34.601-90. Комплекс стандартів на автоматизовані системи. Автоматизовані системи. Стадії створення.

1.4. Планові терміни початку і закінчення роботи зі створення системи

Плановими термінами початку роботи зі створення системи згідно з завданням на дипломне проектування є 1 лютого 2020 року. Планові терміни закінчення роботи зі створення системи з підтримки процесу дослідження задач теорії розкладів, відповідної документації та подання готового програмного продукту – не пізніше 1 червня 2020 року.

					ДП ІС-6223.01.000.ТЗ	Арк.
						4
Змн.	Арк.	№ докум.	Підпис	Дата		

2 ПРИЗНАЧЕННЯ І ЦІЛІ СТВОРЕННЯ СИСТЕМИ

2.1. Призначення системи

Призначенням системи є контролювання продажів та обліку фармацевтичних препаратів, взаємодії працівників та клієнтів аптечних закладів.

2.2. Цілі створення системи

Цілі створення веб-застосування аптечного закладу:

- полегшити взаємодію працівників аптечного закладу та клієнтів
- спростити процес продажу фармацевтичних препаратів.

Для реалізації поставлених цілей система має вирішувати наступні задачі:

- ведення аптечного закладу;
- ведення працівників аптечного закладу;
- ведення електронних рецептів;
- отримання звітності.

3 ХАРАКТЕРИСТИКА ОБ'ЄКТА АВТОМАТИЗАЦІЇ

Об'єктом автоматизації є підсистема продажу та обліку фармацевтичних препаратів корпоративної ERP системи, яка містить великих модулів, а саме:

- модуль для ведення аптечного закладу в електронній системі e-Health;
- модуль для ведення працівників аптечного закладу;
- модуль для ведення особистого облікового запису;
- модуль формування звітів;
- модуль ведення електронних рецептів.

З цією системою можуть працювати працівники будь-якого аптечного закладу.

Робота із системою вимагає особливого технологічного чи технічного забезпечення, оскільки система використовує значну частину оперативної пам'яті обчислювальної системи під час роботи. Детальніша інформація наведена у підрозділі 4.1 та 4.2.

					ДП ІС-6223.01.000.ТЗ	Арк.
						6
Змн.	Арк.	№ докум.	Підпис	Дата		

4 ВИМОГИ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Вимоги до функціональних характеристик

Система має створювати умови для контролю продажів та обліку фармацевтичних препаратів аптечного закладу.

Основні функціональні вимоги до програмного продукту, що були узгоджені з Замовником, наведені нижче.

1. Система надає можливість користувачу змінювати власний профіль.
 - 1.1. Система надає можливість вносити дані власноруч
 - 1.1.1. Система надає можливість змінити пароль
 - 1.1.2. Система надає можливість відновити пароль
2. Система надає можливість директору для ведення аптечного закладу в електронній системі e-Health.
 - 2.1. Система надає можливість директору змінювати дані аптечного закладу.
3. Система надає можливість директору для ведення працівників аптечного закладу.
 - 3.1. Система надає можливість директору реєструвати в системі фармацевтів.
 - 3.2. Система надає можливість директору видаляти з системи фармацевтів.
 - 3.3. Система надає можливість директору редагувати дані фармацевтів.
4. Система надає можливість директору отримувати звіти
 - 4.1. Система надає можливість директору вести діапазон часу, за який сформувати звіт
5. Система надає можливість фармацевту для ведення електронних рецептів
 - 5.1. Система надає можливість фармацевту переглядати інформацію електронних рецептів
 - 5.2. Система надає можливість фармацевту редагувати інформацію електронних рецептів
 - 5.3. Система надає можливість фармацевту створювати погашення на електронний рецепт.

					ДП ІС-6223.01.000.ТЗ	Арк.
						7
Змн.	Арк.	№ докум.	Підпис	Дата		

4.2. Вимоги до надійності

До системи висуваються наступні вимоги надійності:

1. Система має за 1.5 секунди завантажувати усі сторінки програми.
2. Система має надати відповідь користувачу протягом 60 секунд.
 - 2.1. Система має надавати відповідне повідомлення користувачу, якщо потребується більше ніж 60 секунд для обробки інформації
 - 2.2. Система має надавати можливість вибору для користувача для продовження обробки інформації або для відміни у разі того, коли час роботи більше ніж 60 секунд.
 - 2.3. Система має обробляти інформацію у випадку продовження роботи програми.
3. Система має забезпечувати стабільну роботу при помилках користування.
 - 3.1. Система повинна виводити повідомлення про помилку для користувача.
 - 3.2. Система повинна надавати відповідні інструкції користувачу у разі виникнення аварійної ситуації.
 - 3.3. Система повинна відновлювати роботу системи у разі виникнення помилки.

4.3. Умови експлуатації

Для роботи програми необхідні електронно-обчислювальні машини з можливістю стабільної роботи та доступ до мережі Інтернет.

Користувач повинен дотримуватись правил експлуатації системи для забезпечення стабільної та надійної роботи системи.

4.4. Вимоги до складу і параметрів технічних засобів

Для стабільної роботи програми необхідно використовувати технологічні засоби з наступними параметрами:

– персональний комп'ютер у конфігурації: процесор серії Intel i3 CPU m 330 @2.13GHz, не менше 128 Мб оперативної пам'яті, відеоканальний пристрій з розширенням 1024x768, глибина кольору 8 біт;

					ДП ІС-6223.01.000.ТЗ	Арк.
						8
Змн.	Арк.	№ докум.	Підпис	Дата		

- операційна система Windows NT/XP будь-якої версії, які підтримують графічний інтерфейс;
- інтернет-браузер Microsoft Internet Explorer версії 6 та вище або FireFox 2.0 та вище, Google Chrome 11 та вище при використанні ОС Windows.

					ДП ІС-6223.01.000.ТЗ	Арк.
						9
Змн.	Арк.	№ докум.	Підпис	Дата		

5 СТАДІЇ І ЕТАПИ РОЗРОБКИ

Для приведення стадій та етапів розробки підсистеми продажу та обліку фармацевтичних препаратів корпоративної ERP системи наведено календарний план у таблиці 5.1, що попередньо було узгоджено з представниками сторони Замовника.

Таблиця 5.1. – Календарний план виконання робіт

№ з/п	Назва етапів виконання дипломного проекту	Строк виконання
1.	Вивчення рекомендованої літератури	15.04.2019
2.	Аналіз існуючих методів розв'язання задачі	20.04.2019
3.	Постановка та формалізація задачі	27.04.2019
4.	Розробка інформаційного забезпечення	05.04.2019
5.	Алгоритмізація задачі	05.04.2019
6.	Обґрунтування використовуваних технічних засобів	12.04.2019
7.	Розробка програмного забезпечення	06.05.2019
8.	Налагодження програми	07.05.2019
9.	Виконання графічних документів	08.05.2019
10.	Оформлення пояснювальної записки	11.05.2019
11.	Подання ДП на попередній захист	18.05.2019
12.	Подання ДП на основний захист	01.06.2019
13.	Подання ДП рецензенту	02.06.2019

6 ПОРЯДОК КОНТРОЛЮ ТА ПРИЙМАННЯ**6.1. Види випробувань**

Види випробувань та їх зміст узгоджені з представником сторони Замовника.

Для забезпечення якості програмного продукту, повинні бути проведені такі види тестування:

- функціональне;
- модульне;
- інтеграційне;
- системне.

Усі результати проведення тестування мають бути описані у програмі та методиці випробувань дипломного проекту.

					ДП ІС-6223.01.000.ТЗ	Арк.
						11
Змн.	Арк.	№ докум.	Підпис	Дата		

Власник документу:
Попенко Володимир Дмитрович

ID перевірки:
1003979440

Дата перевірки:
12.06.2020 03:32:20 EEST

Тип перевірки:
Doc vs Internet + Library

Дата звіту:
12.06.2020 13:45:05 EEST

ID користувача:
77149

Назва документу: Gaevska_isz61_3

ID файлу: 1003994472 Кількість сторінок: 48 Кількість слів: 5116 Кількість символів: 37667 Розмір файлу: 2.45 MB

11.4% Схожість

Найбільша схожість: 6.72% з джерело https://ela.kpi.ua/bitstream/123456789/30923/1/Braiko_bakalavr.pdf

7.45% Схожість з Інтернет джерелами

12

Page 50

11.4% Текстові збіги по Бібліотеці акаунту

323

Page 50

0% Цитат

Не знайдено жодних цитат

0% Вилучень

Вилучений текст відсутній

Підміна символів

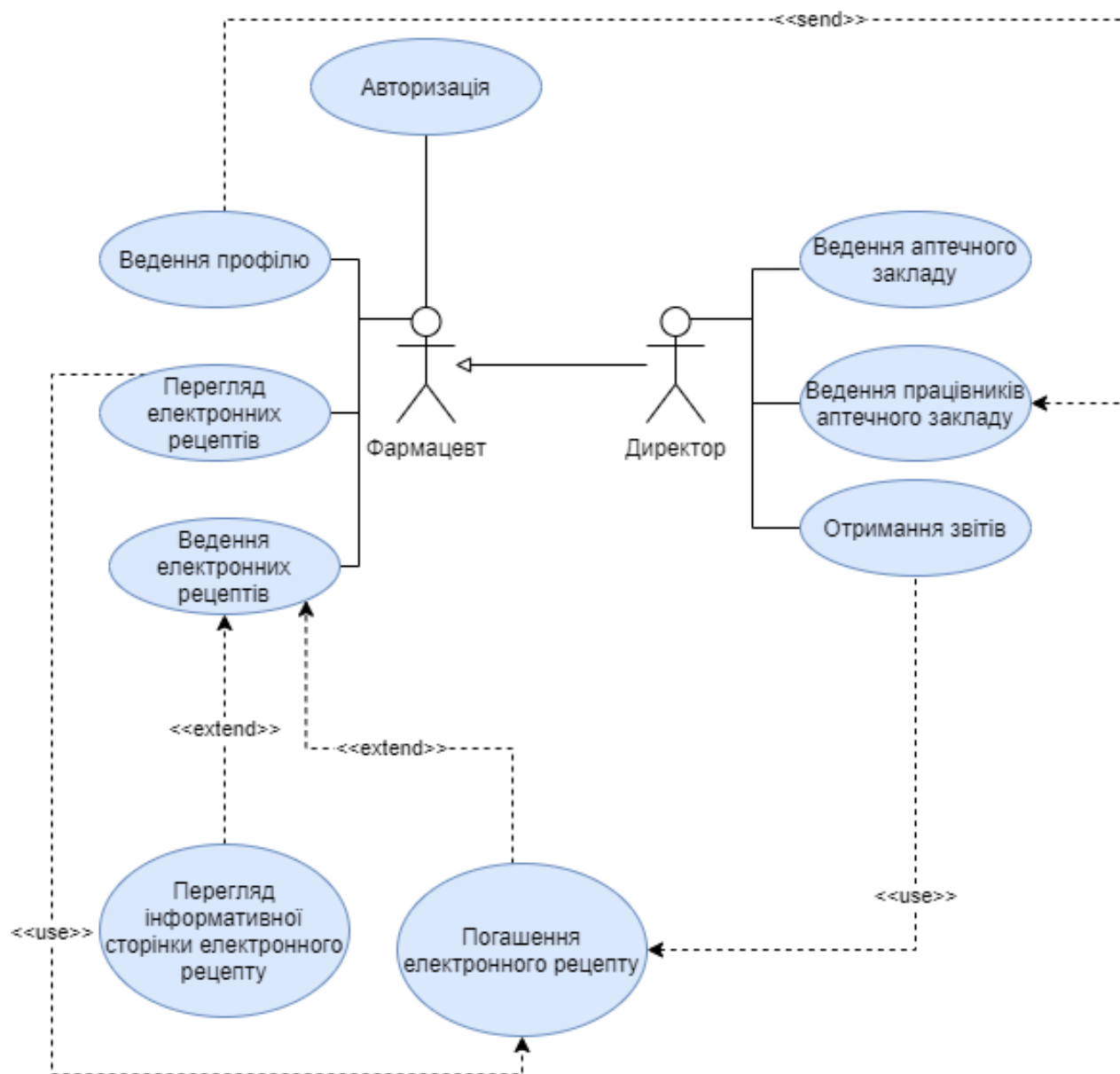
Заміна символів

4

Графічний матеріал до дипломного проєкту

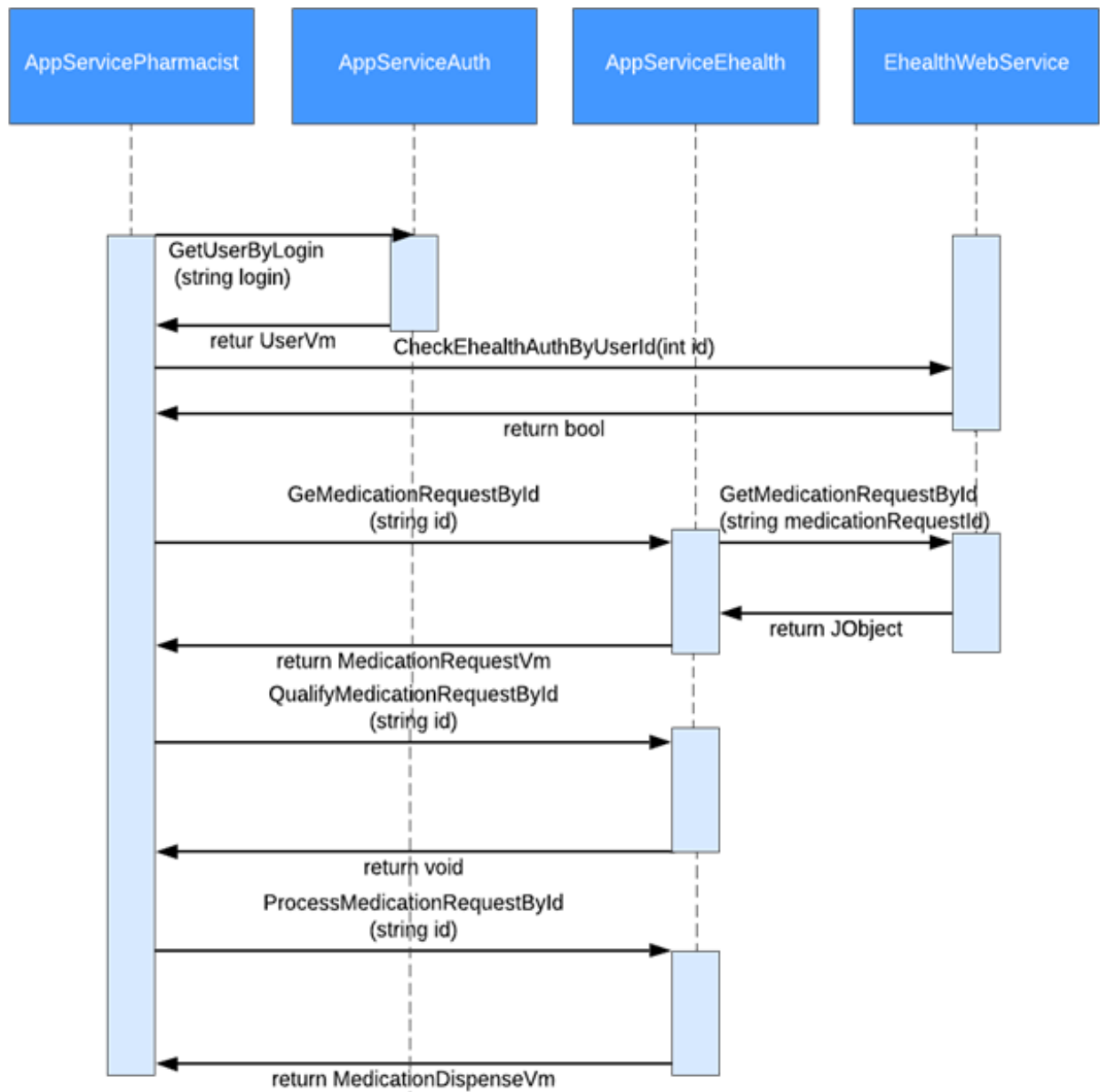
на тему: Підсистема продажу та обліку фармацевтичних препаратів
корпоративної ERP системи

Київ – 2020 року



ДП ІС-6223.02.000.ССВ

					ДП ІС-6223.02.000.ССВ			
					Схема структурна варіантів використання			
Зм.	Арк.	№ документа	Підпис	Дата				
Розробив		Гасвська А.О.			Підсистема продажу та обліку фармацевтичних препаратів корпоративної ERP системи			
Перевірив		Халус О.А.						
Т. кон.								
Н. кон.		Телишева Т.О.			КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-361			
Затвердив		Халус О.А.						



					ДП ІС-6223.03.000.ССП				
					Схема структурна послідовності				
Зм.	Арк.	№ документа	Підпис	Дата	Підсистема продажу та обліку фармацевтичних препаратів корпоративної ERP системи				
Розробив	Гасвська А.О.								
Перевірив	Халус О.А.								
Т. кон.									
Н. кон.	Телишева Т.О.				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-361				
Затвердив	Халус О.А.								

RunedScript	
Id	
Name	
Script	
RunedDate	

_DICT_Genders	
Id	
Name	
Name_en	
Created	
Updated	

_DICT_Roles	
Id	
name	
Created	
Updated	

Configurations	
Id	
Name	
Value	
Description	

Users	
Id	
roleId	
ClinicId	
login	
password	
code	
CodeGeneratedUtc	
[external]	
isLogged	
LastAction	
Created	
Updated	
TabletUserId	

UserToken *	
Id	
UserId	
token	
Created	
Updated	

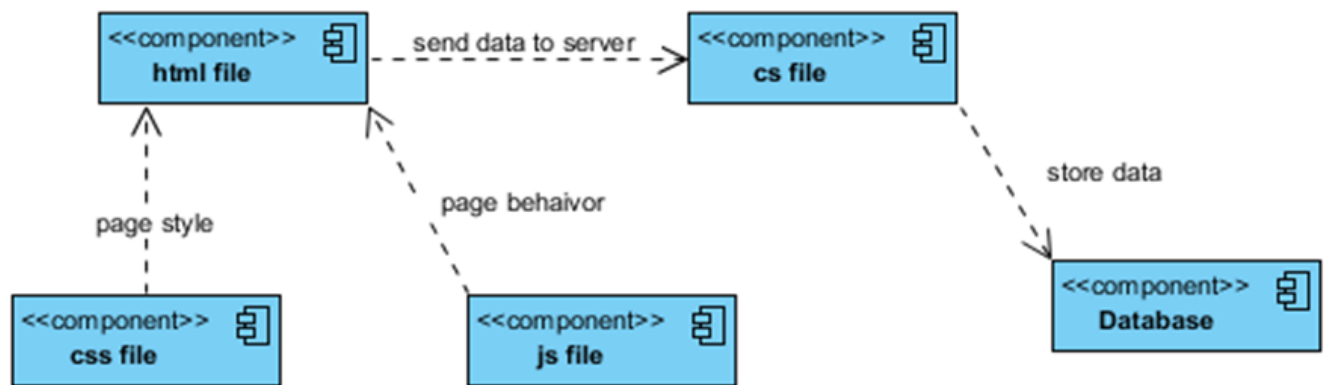
Doctors	
Id	
FirstName	
LastName	
MiddleName	
ClinicId	
Category	
CabinetId	
ImageUrl	
UserId	
Address	
BirthDate	
Email	
Phone	
A1Price	
A2Price	
externalData	
computer	
DismissDate	
ExternalId	
GlobalId	
isOnlineForCalls	
GenderId	
Created	
Updated	
EmailToken	
EmailState	
EmailUpdateTime	
TabletEmployeeId	

EhealthObjectProfile	
Id	
EntryId	
EntryType	
Status	
RegistrationStatus	
JsonData	
Created	
Updated	
EhealthId	
EhealthEntryId	
IsEmailConfirmed	

					ДП ІС-6223.04.000.СБД						
					Схема структурна бази даних						
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Гаєвська А.О.									
Перевірів		Халус О.А.									
Т. кон.					Підсистема продажу та обліку фармацевтичних препаратів корпоративної ERP системи						
Н. кон.		Телишева Т.О.									
Затвердив		Халус О.А.									
					Літера			Маса		Масштаб	
					Аркуш 1			Аркушів 1			
					КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-361						



					ДП ІС-6223.05.000.ССК								
					Схема структурна класів сервісного рівня				Літера		Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата									
Розробив	Гаєвська А.О.												
Перевірів	Халус О.А.								Аркуш 1		Аркушів 1		
Т. кон.					Підсистема продажу та обліку фармацевтичних препаратів корпоративної ERP системи				КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-361				
Н. кон.	Телишева Т.О.												
Затвердив	Халус О.А.												



					ДП ІС-6223.06.000.ССК							
					Схема структурна компонентів	Літера			Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата								
Розробив		Гаєвська А.О.										
Перевірив		Халус О.А.				Аркуш 1			Аркушів 1			
Т. кон.					Підсистема продажу та обліку фармацевтичних препаратів корпоративної ERP системи	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-361						
Н. кон.		Телишева Т.О.										
Затвердив		Халус О.А.										

Вхід до системи

Телефон

Введіть ваш номер телефону

Пароль

Введіть ваш пароль

Я забув(-ла) пароль

Увійти

Заклад

Фармацети

Звіт

Редагувати

ІНФОРМАЦІЯ ПРО АПТЕЧНИЙ ЗАКЛАД

1

ЮРИДИЧНІ ДАНІ

2

ІНФОРМАЦІЯ ПРО АПТЕКУ

3

ІНФОРМАЦІЯ ПРО ДИРЕКТОРА

4

ДАНІ НСЗУ

Назад

Створення картки фармацевта

Завантажити фото

Моб. телефон

+38 (0_) - - -

Прізвище

Прізвище

Ім'я

Ім'я

По батькові

По батькові

Стать

Чоловік

Жінка

Дата народження

Дата народження

Ел. адреса

Ел. адреса

обов'язкові для заповнення поля

Зберегти

Відмінити

Activate Wi
Go to Settings

Пошук рецепта

Показати рецепт

Редагувати

Статус рецепта

Рецепт 0000-6965-5E28-12AT активний

Міжнародна непатентована назва (МНН)

Для виписки рецепта

Лікар

Пацієнт

Лозартан 100 MG таблетки

04.05.2020

Делікатний Фармацевт

Дорошенко В. П.

№90 таблетка

Медичний заклад

Програма

Медичний заклад №1

Доступні ліки

Лікарський засіб

Кількість доз

Роздрібна ціна

Відшкодування за препарат

Доплата пацієнтом

Кількість упаковок

ЛОЗАП® 90 таблетки

90

292,18

156.61

135.57

0

					ДП ІС-6223.07.000.КЕ						
					Креслення вигляду екранних форм.	Літера		Маса		Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Гаєвська А.О.									
Перевірів		Халус О.А.									
Т. кон.						Аркуш 1			Аркушів 1		
					Підсистема продажу та обліку фармацевтичних препаратів корпоративної ERP системи	КПІ ім. Ігоря Сікорського кафедра АСОІУ гр. ІС-361					
Н. кон.		Телишева Т.О.									
Затвердив		Халус О.А.									